Journal of Building Engineering 63 (2023) 105493

Contents lists available at ScienceDirect

Journal of Building Engineering

journal homepage: www.elsevier.com/locate/jobe

An interpretable machine learning method for the prediction of R/C buildings' seismic response

Konstantinos Demertzis [a,b], Konstantinos Kostinakis [c,*], Konstantinos Morfidis [d], Lazaros Iliadis [b]

[a] School of Science & Technology, Informatics Studies, Hellenic Open University, Greece
[b] School of Engineering, Department of Civil Engineering, Faculty of Mathematics Programming and General Courses, Democritus University of Thrace, Kimmeria, Xanthi, Greece
[c] Department of Civil Engineering, Aristotle University of Thessaloniki, Aristotle University Campus, 54124, Thessaloniki, Greece
[d] Earthquake Planning and Protection Organization (EPPO-ITSAK), Terma Dasylliou, 55535, Thessaloniki, Greece

ARTICLE INFO

Keywords:
Interpretable machine learning
Model validation
Seismic damage prediction
Structural vulnerability assessment
Reinforced concrete buildings

ABSTRACT

Building seismic assessment is at the forefront of modern scientific research. Several researchers have proposed methods for estimating the damage response of buildings subjected to earthquake motions without conducting time-consuming analyses. The advancement of computer power has resulted in the development of modern soft computing methods based on the use of Machine Learning (ML) algorithms. However, a lack of expertise associated with the use of complex ML architectures can affect the performance of the intelligent model and, ultimately, reduce the algorithm's reliability and generalization which should characterize these systems. The current paper proposes a fully validated interpretable ML method for predicting seismic damage of R/C buildings. Specifically, the most efficient machine learning algorithms were used in a large-scale comparison study in a sophisticated dataset of 3D R/C buildings. Moreover, effective additional validation ensures that models are sound, have low complexity, are fair and provide clear explanations for decisions made. Also, extensive experiments were done to make the final machine learning model explainable and the decisions interpretable. The proposed method aims to suggest that the civil protection mechanisms must include scientific methodology and appropriate technical tools into their technological systems, in order to make substantial innovative leaps in the new era.

1. Introduction

One of the most important, but also challenging, scientific issues in the field of earthquake engineering is the estimation of the structural response of buildings subjected to earthquake ground motions. Since now, numerous research studies have dealt with the above issue and proposed a vast variety of different methods aiming at the seismic assessment of structures. Many of these methods focus on the rapid determination of the earthquake damage response and on the seismic vulnerability assessment of large number of buildings without performing computationally hard analyses, in an attempt to overcome the difficulties resulting from the time-consuming conduction of demanding nonlinear analysis methods (e.g Refs. [1–6]), These procedures usually utilize methods based on the application of statistics theory. In the last decades, the increase of the computers' power has led to the development of modern

* Corresponding author.
E-mail address: kkostina@civil.auth.gr (K. Kostinakis).

https://doi.org/10.1016/j.jobe.2022.105493
Received 21 July 2022; Received in revised form 13 October 2022; Accepted 29 October 2022
Available online 4 November 2022

statistical methods based on the adoption of Machine Learning (ML) algorithms. The up-to-date research on these methods revealed that they can provide a fast, reliable, and computationally easy way for screening of vulnerable structures and that they can be used as an efficient alternative to the conduction of demanding numerical simulations (e.g. Refs. [7–12]). The achievement of this goal is made through the creation of a relationship mapping that emulates the structure's behavior.

ML is one of the most important scientific fields of the new era that includes those algorithmic methods that can be learned from data. Data-driven intelligent systems can translate human knowledge and experience into right and timely decisions. They combine ideas from the sciences of statistics and probabilities to make accurate future predictions, while mathematical optimization techniques are used to improve the performance of a system. There are four distinct categories of ML with independent characteristics of learning: a) the information-based learning methodologies that employ concepts from information theory to build models (e.g. Decision Tree Algorithm), b) the similarity-based learning methods that build models based on comparing features of known and unknown objects or measure similarity between past and forthcoming occurrences (e.g. k-Nearest Neighbor), c) the probability-based learning techniques that build models based on measuring how likely it is that some event will occur (e.g. Naive Bayes Classifier) and, finally d) the error-based learning that builds models based on minimizing the total error through a set of training instances (e.g. Support Vector Machine). On the other hand, based on how to use the data, there are three main categories of ML algorithms: a) Supervised Learning in which the training process of the algorithm is based on samples of labeled data, b) Unsupervised Learning which is the ability of the algorithm to detect patterns in unknown data and, finally, c) Reinforcement Learning that employs algorithms for discovering the environment based on rewarded actions.

Several research studies have proved that the ML methods, mainly Artificial Neural Networks (ANNs), can effectively assess the seismic response of complex structures. A comprehensive literature review of the most commonly used and newly developed ML techniques for the assessment of the buildings' damage has been made by Harirchian et al. [13], by Xie et al. [14] and by Sun et al. [15]. A brief review of some of the most important research works is given below. Latour and Omenzetter [16] were among the first researchers who studied the ability of the ANNs to reliably estimate the earthquake-induced damage of planar R/C frames by using nonlinear time history analyses' results. A similar investigation was carried out by Arslan [17], who studied the impact of certain structural parameters on the damage level of regular R/C buildings under seismic ground motions. Kia and Sensoy [18] investigated the impact of certain seismic parameters on the ability of ANNs to assess the seismic damage level of R/C concrete frames based on nonlinear time history analyses of a 2D moment resisting R/C frame. Kostinakis and Morfidis conducted a series of research studies [19–21] in an attempt to estimate the reliability of ANNs as regards the estimation of the seismic response of R/C buildings. More recently, Zhang et al. [22] proposed a ML framework for the assessment of the post-earthquake structural safety of a 4-story R/C special moment frame building. In another research conducted by the same research team [23] several ML methods were utilized in order to adequately estimate the residual structural capacity of damaged tall buildings.

The results of most research studies established the ability of ML techniques to successfully solve engineering problems. However, all of the abovementioned researchers adopted black box ML methods for their research; namely, no study has attempted to utilize explainable techniques in order to evaluate their efficiency in assessing the damage response with adequate reliability. The present paper aims to evaluate extensively many Machine Learning algorithms for the reliable prediction of 3D R/C buildings' seismic response. Moreover, for the winner algorithm, which produced training stability, high overall performance and a great generalization to estimate the ability to predict the damage response of buildings, a large-scale validation was made. Exact model validation is the best technique for mitigating the effects of this difficulty. This additional validation step checks the model's predictions with an unknown/unseen dataset to assess the model's predictive capacity and performance and, consequently, to increase the model's output reliability. Note that this technique avoids the potential that the machine learning models would exhibit unexpected behaviors in response to random input data or malicious attacks, because the different validation phase analyzes the model to find weaknesses and evaluate its resilience. Also, extensive experiments were done in order to make the final machine learning model explainable and the decisions interpretable.

2. Formulation of the problem in terms compatible to machine learning methods

2.1. Overview of the procedure

In this section the procedure adopted in order to formulate the problem in terms compatible to ML methods is presented. The procedure consists of the following steps:

- Generation of the training dataset, which includes selection of a large number of representative R/C buildings, design and modeling of the inelastic properties of the them and selection of an adequate number of seismic motions.
- Selection of the problem's input (structural and seismic) parameters.
- Conduction of Nonlinear Time History Analyses (NTHA), according to which the buildings are analyzed for the selected earthquake records and their seismic response is determined. Consequently, processing of the analyses' results in order to compute the values of an appropriate seismic damage index (in the present study the MIDR (Maximum Interstory Drift Ratio) index), which is selected as the output parameter (target) of the ML procedures.

2.2. Training dataset

In order to fulfill the purposes of the present research study, a large training dataset consisting of buildings with a variety of structural characteristics was considered. An attempt was made to select structures that are representative of the buildings designed and built with the aid of modern seismic codes and according to the common construction practice in European countries with regions

of high seismicity. More specifically, a set of 30 R/C buildings was selected [20,21]. The buildings' structural system consists of members in two perpendicular directions (denoted as axes x and y). Moreover, they are rectangular in plan and regular in elevation and in plan according to the criteria set by EN1998-1 [24]. The buildings possess different characteristics concerning the stories' number n_{st} (stories' height: 3.2 m), the value of structural eccentricity e_o (i.e. the distance between the mass center and the stiffness center of stories) and the ratio of the base shear received by the walls along two horizontal orthogonal directions (axes x and y): V_{w1} and V_{w2}. The values of these structural parameters for the selected buildings are given in Table 1 (more details about the selected buildings can be found in Refs. [20,21]).

In the above table, L_x and L_y are the dimensions of the rectangular shaped plans of the selected buildings and $e_0=(e_{0x}^2+e_{0y}^2)^{1/2}$, where e_{0x}, e_{0y} are the structural eccentricities along axes x and y respectively.

In order to investigate the impact of the masonry infills on the seismic response and damage of the buildings, for each one of the 30 structures three different assumptions about the distribution of the masonry infills were considered, leading to three different training subsets: (a) subset denoted as ROW_FORM_BARE consisting of the 30 buildings without masonry infills (bare structures), (b) subset denoted as ROW_FORM_FULL-MASONRY consisting of the 30 buildings with masonry infills uniformly distributed along the height (infilled structures) and (c) subset denoted as ROW_FORM_PILOTIS consisting of the 30 buildings with the first story bare and the upper stories infilled (buildings with pilotis). Consequently, the total number of buildings investigated herein is 30 different structural systems x 3 different distributions of masonry infills = 90. The three abovementioned subsets of the buildings, as a result of their different masonry infills' configurations, were trained separately by the same ML methods, in order to draw conclusions about the possible differences in the predictive ability of the ML techniques, resulting from the influence of the infill walls on the seismic response of them.

The 30 selected bare buildings were modeled, analyzed and designed according to the provisions of EN1992-1-1 [25] and EN1998-1 [24]. For the buildings' elastic modelling all recommendations of EN1998-1 were followed (diaphragmatic behavior of the slabs, rigid zones in the joint regions of beams/columns and beams/walls, values of flexural and shear stiffness corresponding to cracked R/C elements). The buildings were classified as Medium Ductility Class (MDC) structures. The analyses and design were done with the aid of the modal response spectrum method, as defined in EN1998-1. All buildings were designed for the combination of vertical loads 1.35G+1.50Q, as well as the seismic combination G+0.3Q ± E, (where G, Q are the dead and live loads, and E is the seismic action expressed by the simultaneous application of the design spectrum of EN1998-1 along the direction of axes x and y). The design of the structural members was made following the provisions of EN1992-1-1 and EN1998-1, utilizing the professional program for R/C building analysis and design RAF [26].

After the elastic modeling and design of the bare buildings, the three subsets mentioned above (bare buildings, infilled buildings, buildings with pilotis) were created and their nonlinear behavior was simulated, in order to analyze them by means of NTHA. The modeling of the structures' nonlinear behavior was made using lumped plasticity models (plastic hinges at the column and beam ends, as well as at the base of the walls). The Modified Takeda hysteresis rule [27] was adopted in order to model the material inelasticity of the structural members. Moreover, the effects of axial load-biaxial bending moments (P-M_1-M_2) interaction at columns and walls hinges were taken into consideration. The yield moments of the R/C elements and the parameters which were necessary for the determination of the P-M_1-M_2 interaction diagram of the vertical R/C elements' cross sections were computed using the XTRACT software [28].

Regarding the infill walls' modeling, in the present study, the equivalent diagonal strut model was adopted. This model is one of the most well-known and documented in the relevant literature macro-models [29,30]. It does not account for the local failure, but it participates in the building's global collapse mechanism, which is the main objective of the present study. In particular, each infill panel was modeled as single equivalent diagonal strut with stress-strain diagram according to the model proposed by Crisafulli [31] (Fig. 1). Fig. 1 illustrates the simulation of the masonry infills based on the Crisafulli model, along with all the basic parameters used to define the properties of the diagonal struts. Note that the values of these parameters were computed with the aid of the code provisions given in EN1996-1-1 [32].

Table 1
The values of structural parameters of the selected R/C buildings.

No.	n_{st}	L_x(m)	L_y(m)	e_o(m)	V_{w1} (%)	V_{w2} (%)	No.	n_{st}	L_x(m)	L_y(m)	e_o(m)	V_{w1} (%)	V_{w2} (%)
1	3	13.5	10.0	0.0	0.0	0	16	3	13.0	9.0	0.98	0.0	0.0
2	5	20.0	14.0	0.0	0.0	0.0	17	5	17.5	10.0	2.58	0.0	0.0
3	7	20.0	14.0	0.0	0.0	0.0	18	7	17.5	10.0	2.39	0.0	0.0
4	3	15.0	10.0	0.0	73.0	76.0	19	3	13.5	9.0	4.65	52.0	46.0
5	5	19.0	16.2	0.0	77.0	80.0	20	5	16.0	14.5	4.19	43.0	42.0
6	7	19.0	16.2	0.0	57.0	64.0	21	7	16.0	14.5	3.79	37.0	36.0
7	3	15.0	15.0	0.0	41.0	41.0	22	3	13.5	9.0	2.23	47.0	0.0
8	5	21.2	18.7	0.0	46.0	50.0	23	5	16.0	14.5	2.65	38.0	0.0
9	7	21.2	18.7	0.0	43.0	46.0	24	7	16.0	14.5	2.49	35.0	0.0
10	3	17.0	12.5	0.0	43.0	0.0	25	3	14.5	9.0	3.53	64.0	0.0
11	5	20.2	15.2	0.0	41.0	0.0	26	5	14.0	16.0	3.01	0.0	69.0
12	7	20.2	15.2	0.0	38.0	0.0	27	7	14.0	16.0	3.01	0.0	65.0
13	3	15.0	10.0	0.0	77.0	0.0	28	3	13.5	10.0	6.73	64.0	58.0
14	5	20.2	15.2	0.0	68.0	0.0	29	5	16.5	16.5	6.29	65.0	72.0
15	7	20.2	15.2	0.0	51.0	0.0	30	7	16.5	16.5	5.96	59.0	67.0

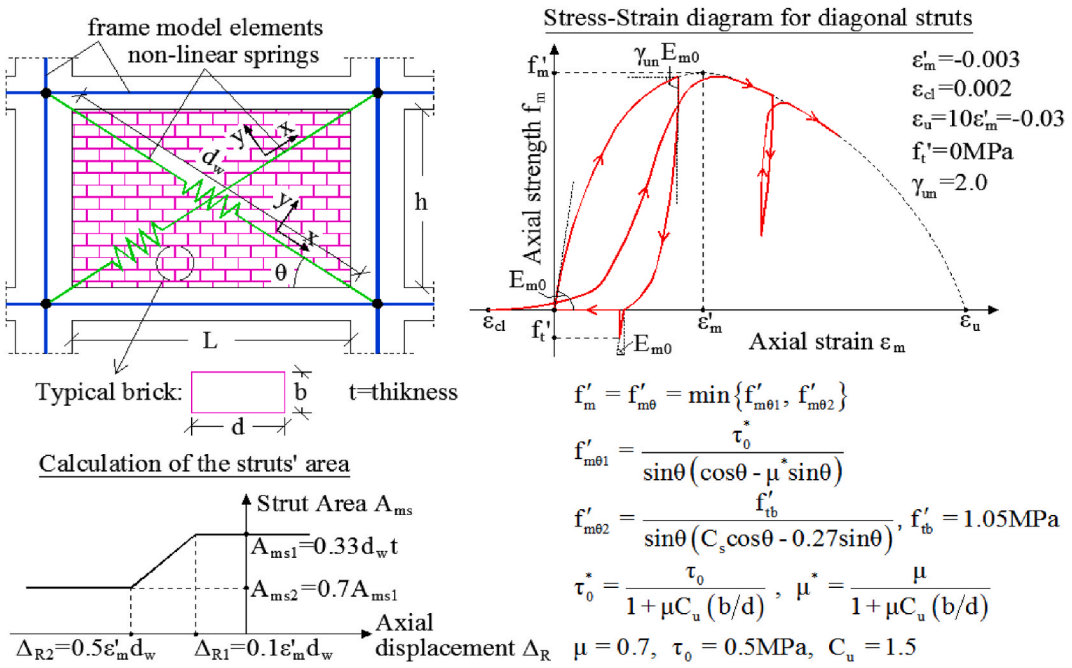

Fig. 1. Simulation of the masonry infill response using the method of diagonal struts.

2.3. Input parameters

The Machine Learning methods are computational structures which are capable of approaching the solution of multi-parametric problems. This feature gives the flexibility to select the number of the parameters (input parameters) through which a problem can be formulated. For the present investigation's purposes, both structural and seismic parameters were chosen in order to adequately describe the problem. Considering the structural parameters, four macroscopic characteristic, which are considered crucial for the vulnerability assessment of existing 3D R/C buildings were selected: the total height of buildings H_{tot}, the ratios of the base shear that is received by R/C walls (if they exist) along two horizontal orthogonal directions x and y (ratio V_{w1} and ratio V_{w2}) and the structural eccentricity e_0 (Table 1). The above selected structural parameters are widely used in well-known methods of seismic vulnerability assessment of existing R/C buildings (e.g. Refs. [2,5]). They have also been recognized by the modern seismic codes as the parameters which have significant effect on R/C buildings seismic response (e.g. Ref. [24]). As regards the seismic parameters, it must be noticed that there are many definitions of them, which are obtained from the accelerograms records. For the present study, the 14 seismic parameters presented in Table 2 have been chosen [33,34], in an attempt to select the ones widely used by the relevant literature to describe better the seismic excitations and their impact to structures. The significantly larger number of seismic parameters selected (14) compared to the 4 structural parameters was based on the fact that the uncertainty associated with the seismic excitations is much larger than the uncertainty associated with the estimation of the structural parameters. Thus, the large number of selected seismic parameters, in combination with the selection of the 65 seismic excitations (Appendix A), has as a result to cover a wide range of values of these parameters, ensuring the effective performance of the examined machine learning methods.

Table 2
The selected seismic (ground motion) parameters and the ranges of their values corresponding to the 65 earthquakes.

Ground Motion Parameter	Minimum Value	Maximum Value
Peak Ground Acceleration - PGA	0.004 g	0.822 g
Peak Ground Velocity - PGV	0.86 cm/s	99.35 cm/s
Peak Ground Displacement - PGD	0.36 cm	60.19 cm
Arias Intensity **I_a**	≈0.0 m/s	5.592 m/s
Specific Energy Density - SED	1.24 cm^2/s	16762.8 cm^2/s
Cumulative Absolute Velocity - CAV	14.67 cm/s	2684.1 cm/s
Acceleration Spectrum Intensity - ASI	0.003 g s	0.633 g s
Housner Intensity - HI	3.94 cm	317.6 cm
Effective Peak Acceleration - EPA	0.003 g	0.63 g
V_{max}/A_{max} (PGV/PGA)	0.036 sec	0.336 s
Predominant Period - PP	0.077 s	1.26 s
Uniform Duration - UD	≈0.0 s	17.68 s
Bracketed Duration - BD	≈0.0 s	61.87 s
Significant Duration - SD	1.74 s	50.98 s

2.4. Output parameters - targets

The exported result of the solution of the problem which is examined in the present paper is the estimation of the seismic damage state of R/C buildings, so a reliable measure that can adequately quantify their damage response must be adopted as a target (output parameter) for the Machine Learning algorithms. More specifically, the 90 R/C buildings presented above were analyzed by means of NTHA for a suite of 65 earthquake ground motions, accounting for the design vertical loads. As a consequence, a total of 5850 NTHA (90 buildings x 65 earthquake records) were conducted in the present research. The analyses were performed using the computer program Ruaumoko [35]. It must be mentioned that the conduction of such a large number of nonlinear analyses was a rather computationally-challenging task, so in order to accomplish it, a series of scripts was constructed using the Visual Basic programming language. Regarding the selection of the input earthquake motions, each of these consists of a pair of horizontal bidirectional seismic components, obtained from the PEER [36] and the European strong-Motion database [37]. The selection of the records was made bearing in mind the coverage of a large variety of realistic values for the 14 ground motion parameters considered as inputs. In Table 2 the range of the ground motion parameters' values that correspond to the 65 chosen strong motions is depicted. For the calculation of the above seismic parameters, the computer program SeismoSignal [34] was utilized.

For each one of the nonlinear analyses, the assessment of the seismic damage was determined. In particular, the estimation of the seismic damages that are expected to occur in structural members of R/C buildings is accomplished through the calculation of certain measures which try to quantify the severity of the damage. The choice of a reliable damage measure, that can adequately capture the damage level of the building, is a very difficult task, since it depends on numerous parameters. The present research study, in order to express the buildings' seismic damage, adopts the Maximum Interstory Drift Ratio (MIDR). More specifically, MIDR corresponds to the maximum story's drift among the perimeter frames and it is calculated according to Fig. 2. The MIDR, which is extensively used as an effective indicator of structural and nonstructural damage of R/C buildings (e.g. Refs. [38,39]), has been adopted by many researchers for the assessment of the structures' inelastic response.

3. Presentation of used machine learning algorithms

A lack of technical expertise and experience associated with the usage of complicated machine learning architectures, on the other hand, might influence the performance of the machine learning models and impede the proper design of some crucial hyper-parameters. These potential misconfigurations decrease the algorithm's dependability and generalizability in these systems. In order to identify the most effective algorithm that is capable to predict the R/C buildings' seismic damage with high accuracy, an extensive comparison with the most widely used supervised ML models was made. A comprehensive review of the comparison models is summarized as follows:

1. **Light Gradient Boosting Machine:** is a gradient boosting framework based on decision trees to increases the efficiency of the model and reduces memory usage [40].
2. **Gradient Boosting Regressor:** This method produces an ensemble prediction model by a set of weak decision trees prediction models. It builds the model smoothly, allowing at the same time the optimization of an arbitrarily differentiable loss function [41].
3. **Random Forest Regressor:** A Random Forest is a meta-learner that builds a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and to control over-fitting [42].
4. **Extra Trees Regressor:** Extra Trees is an information-based learning methodology. Specifically, it is an ensemble machine learning algorithm that combines the predictions from many decision trees [43].
5. **k-Nearest Neighbors Regressor:** k-Nearest Neighbors Regressor is a similarity-based learning algorithm, according to which the target is predicted by local interpolation of the targets associated with the nearest neighbors in the training set [44].
6. **Linear Regression:** Linear Regression is a model that assumes a linear relationship between the input variables (x) and the output variable (y), so that (y) can be calculated from a linear combination of the input variables (x). In linear regression, relationships are modeled using linear prediction functions whose unknown model parameters are estimated from the probability distribution of the prediction values [45].
7. **Bayesian Ridge:** Bayesian Ridge is a type of linear regression algorithm that uses probability distributions rather than point estimates in order to solve a regression problem [46].

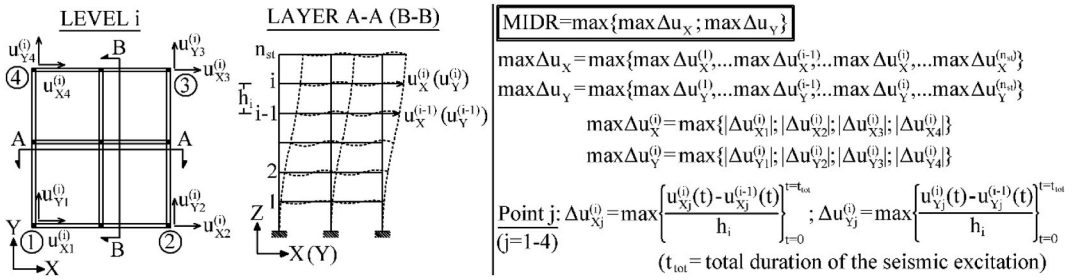

Fig. 2. Determination of the MIDR in the case of a n-story 3-D building with arbitrary plan-view.

8. **Ridge Regression:** Ridge Regression is a regression method that does not provide confidence limits. It uses regularization L2-norm in order to solve a high covariance problem, even if the errors come from an abnormal distribution [47].
9. **Decision Tree Regressor**: A decision tree is a tree-based model including chance event outcomes, resource costs, in order to displays conditional control statements. Each node represents an attribute, each branch represents the outcome of an attribute test, and each leaf represents the decision taken after computing all attributes. The paths from the root to leaf represent the regression process [48].
10. **AdaBoost Regressor:** It is a meta-learner that begins by fitting a regressor on the original dataset and then fits additional copies of the regressor on the same dataset where the weights of instances are adjusted according to the error of the current prediction [49].
11. **Elastic Net:** The Elastic Net is a normalized regression method to fit data that linearly combines the L1 and L2 norms of the lasso and ridge regression methods [50].
12. **Lasso Regression:** Least Absolute Shrinkage and Selection Operator Lasso Regression is a type of linear regression methodology that uses a shrinkage technique in which data are shrunk to a central point, such as the average value [51].
13. **Orthogonal Matching Pursuit:** Orthogonal Matching Pursuit is a sparse approximation algorithm which finds the optimal multidimensional data projection fitting the data with high accuracy [52].
14. **Huber Regressor:** Huber Regressor is a regression method which defines a threshold based on the distance between target and prediction that makes the loss function switch from a squared error to an absolute one [53].
15. **Least Angle Regression:** Least Angle Regression is a linear regression algorithm for fitting high-dimensional data. The solution consists of a curve denoting the solution for each value of the L1 norm of the parameter vector in which the estimated parameters are increased in a direction equiangular to each one's correlations with the residual [54].

4. Comparative assessment of the ML methods

The abovementioned ML techniques were utilized for the statistical analysis of the training datasets in order to estimate their predictability in the estimation of the buildings' seismic damage. The following regression metrics were used to compare the results and to detect the ML algorithm which is the most efficient:

Coefficient of Determination - . In order to express the correlation between two random variables, R^2 is used which is expressed in terms of percentage. This metric gives the rate of variability of the Y values calculated by X and vice versa. R^2 is defined as follows:

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

where Y_i are the observed values of the dependent variable, $\widehat{Y}_i$ are the estimated values of the dependent variable, $\overline{Y}$ is the arithmetic mean of the observed values and n is the number of observations. R^2 attains values in the interval [0,1], with optimal performance when its values approach the unit, indicating that the regression model adapts optimally to the data.

Mean Absolute Error – MAE. MAE is the measure that quantifies the error between the estimated and the observed values. It is calculated by the formula:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |f_i - y_i| = \frac{1}{n} \sum_{i=1}^{n} |e_i| \tag{2}$$

where f_i is the estimated values and y_i is the observed ones. The average of the absolute value of the difference between these values is defined as the absolute error of their relation $|e_i| = |f_i - y_i|$.

Mean Square Error – MSE. MSE is the basic comparison measure that calculates how well a model approaches the number of control examples in a regression process. It is given by the following formula:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (\widehat{Y}_i - Y_i)^2 \tag{3}$$

where Y is an observed value and $\widehat{Y}$ is an estimated value for the n predictions.

Root Mean Squared Error – RMSE. RMSE calculates the average error of the predicted values in relation to the actual values. RMSE is based on the following formula:

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^{n} \left(P_{(ij)} - T_j\right)^2} \tag{4}$$

where $P_{(ij)}$ is the value predicted by program i for a simple hypothesis j and T_j is the target value for the simple hypothesis j. The success of a regression model requires extremely small values for the RMSE, while the best case (absolute correlation between actual and predicted values and therefore absolute success of the model) is achieved when $P_{(ij)} - T_j = 0$.

Mean Absolute Percentage Error – MAPE. MAPE provides an objective measure of the estimation error as a percentage of demand (e.g. the estimation error is on average 10% of actual demand) without depending on the order of magnitude of demand. It is given by the following formula where A_t is the actual value and F_t is the forecast value:

$$MAPE = 100 \sum_{t=1}^{T} \frac{\left[\frac{|A_t - F_t|}{A_t}\right]}{T} \tag{5}$$

Generally speaking, RMSE gives more importance to the highest errors, hence it is more sensitive to outliers, whereas, on the other hand, MAE is more robust to outliers. RMSE and MSE work on the principle of averaging the errors, while MAE's calculation is based on the median of the error. Finally, MAPE is a very intuitive interpretation in terms of relative error.

In order to confirm the effectiveness of the ML algorithms, extensive ML tests were performed and the comparative results (ranked form the most efficient to the least efficient method) obtained for each one of the three datasets in terms of the abovementioned metrics are presented in the Tables given in Appendix B. Tables B1, B2, and B3 clearly show the superiority of the Light Gradient Boosting Machine (LightGBM) algorithm, which excels in all metrics, while the performance error remains very low compared to the other approaches. Specifically, the accuracy of the LightGBM, exceeds on average the second-best method by almost 3.5%, while the recorded error is significantly smaller. These features are clearly demonstrated by the very high-performance results that it has achieved, as well as its ability to generalize to new unknown situations and to effectively model real-world data. Specifically, the results revealed that using LightGBM it is possible to correlate sophisticated parameters in a simple way and to solve dynamic problems like the prediction of the R/C buildings' seismic response with high accuracy and with an affordable computational cost.

5. Description of the implementation of the most efficient algorithm: LightGBM

In the following, a thorough description, along with analytical details, of the implementation of the most efficient ML algorithm (LightGBM) are given. LightGBM [40] is an information-based learning methodology, which belongs to the class of gradient boosting algorithms and uses a learning algorithm based on regression trees. Regression trees are a simple, easy-to-interpret technique that works best in single-dimensional data analysis (not multidimensional data such as photos, videos, etc.). Considering a set of the form (x_i, y_i) for $i = 1, 2, \ldots, N$ with $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ and for $j = 1, 2, \ldots, p$, the construction of a regression tree is defined as follows:

1. The set of target variable's values y_i is divided into M regions $R_1, R_2, \ldots, R_M$
2. The variable is modeled as a constant c_m in each region so that:

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m) \tag{6}$$

Having as a criterion of minimization the sum of the squares $\sum (y_i - f(x_i))^2$ it is easy to calculate the optimal $\widehat{c}_m$, which is the average of y_i in the region m:

$$c_m = ave(x_i | y_i \in R_m) \tag{7}$$

The problem which arises is that using the sum of the squares in order to find the best results, the algorithm becomes extremely time-consuming. For this reason, another approach is usually used, according to which in each step the target variable is divided into two areas through two branches, a variable X_j and the separation point s are selected, which results in the largest reduction in the sum of squares. Essentially, in this way a variable j and a point s are sought, in order to minimize the following function:

$$\sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2 \tag{8}$$

where $R_1(j,s) = \{X | X\_j \leq s\}$ και $R_2(j,s) = \{X | X\_j > s\}$. Then, the process is repeated for each area created. The question that arises is how big the trees should be. Note that a large tree will be very specialized in data resulting in a low predictive ability for new data that they have never seen before, while a small tree may not have been properly trained resulting in yielding unsatisfactory results. One solution to the problem is to set a minimum threshold and only if the reduction in the sum of squares achieved by the division is larger than the threshold the separation takes place. This strategy is not always optimal, as a bad initial separation can then lead to a very good next one. The strategy that works best is pruning the tree. The idea is to grow a tree with a predetermined number of nodes and then to prune it using a criterion based on the complexity of the tree as follows:

1. Firstly, a tree is trained at least $T \subset T_0$, which can be any tree that resulted from the pruning of the tree T_0.
2. Setting the terminal nodes of T, with the node m representing the region R_m, then:

$$\widehat{c}_m = \frac{1}{N_m} \sum_{x_i \in R_m} y_i \tag{9}$$

$$Q_m(T) = \sum_{x_i \in R_m} (y_i - \widehat{c}_m)^2 \tag{10}$$

$$Ca(T) = \sum_{m=1}^{|T|} Q_m(T) + a|T| \tag{11}$$

Essentially, the first term of the function Ca measures how well the tree adapts to the training data (small values indicate good adaptation) and the second term measures the complexity of the tree. The parameter $\alpha \geq 0$ indicates the counterpoint between complexity and good fit of the tree. For $\alpha = 0$ the resulting tree is T_0, as no cost is added for each node included in the tree. As the parameter α grows, the cost of the tree complexity increases, so it results in smaller trees which do not adapt as well to the training data. The smaller the parameter α the larger the tree that is constructed, resulting often in overfitting in the training data and, consequently, in a poor performance for other datasets.

As mentioned above, LightGBM is a gradient boosting algorithm. The Boosting technique is based on the creation of successive trees. Each tree is trained using information from previous trees. The algorithm works as follows:

1. For each observation in the set of training data $\hat{f}(x) = 0$ and $\varepsilon_i = y_i$ is set.
2. In each round k a tree $\widehat{f^k}$ with d nodes is trained, having as a response variable the residuals of the operation (what is left over from the previous regression round) which are denoted by ε_i.
3. A pruned version of the new tree is added so that:

$$\widehat{f}(x) \leftarrow \widehat{f}(x) + \lambda \widehat{f^k}(x) \tag{12}$$

4 Respectively:

$$\varepsilon_i \leftarrow \varepsilon_i - \lambda \widehat{f^k}(x) \tag{13}$$

5. Repeating the process from step 2 for K times (K is defined by the user) the final form of the model is obtained:

$$\widehat{f}(x) = \lambda \sum_{k=1}^{K} \widehat{f^k}(x) \tag{14}$$

In order the Boosting technique to be effective, the user must specify the number of trees to be created, the parameter λ and the number of nodes in each tree. A large number of trees can easily be over-adapted to training data resulting in a poor generalization ability. The λ parameter determines how fast the model will learn. Typical values of λ are from 0.001 to 0.1. The number of nodes controls the complexity of each tree. Often, trees of a single division, also known as branches, are satisfactory because the learning in the model is done slowly and in a controlled way.

The Gradient Boosting technique is an extension of the Boosting technique, combining two methods, the Gradient Descent algorithm and the Boosting technique. Gradient Descent is a first-class optimization method. In order to find the total minimum of a function using this technique, its derivative is firstly calculated and then the inverse process of finding the derivative is used. The derivative measures how much the value of a function $J(\theta)$ will change if the variable θ changes slightly. It is essentially the slope of the function. High values of the function indicate a large slope and therefore a large change in the value of (θ) for small changes of θ. This algorithm is iterative, namely it initializes a random value in θ, calculates the derivative of the function at the given point and modifies θ so that:

$$\theta = \theta - \rho \frac{dj}{d\theta} \tag{15}$$

where the parameter ρ determines how fast it will move in the negative direction of the derivative. The process is repeated until the algorithm converges.

In the case of Gradient Boosting, the algorithm suggests training trees in the negative derivative of the loss function. For example, taking as a loss function the sum of the squares of the residuals ε_i divided by 2 so that:

$$L(y_i, \widehat{y}_i) = \frac{1}{2} \sum_{i=1}^{N} (y_i - \widehat{y}_i)^2 \tag{16}$$

Calculating the derivative:

$$\frac{dL(y_i, \widehat{y}_i)}{d\widehat{y}_i} = \widehat{y}_i - y_i \tag{17}$$

That is, the negative derivative of the loss function equals to the residuals ε_i. So, essentially, the process involves training a tree based on the ε_i residuals, to which a pruned by ρ version of the new tree is added. In this way, the Gradient Boosting technique adds successive trees at any given time t to the negative derivative of the loss function so that:

$$\widehat{y}_i^{(t)} = \sum_{t=1}^{K} f_t(x_i),\ f_t \in F \tag{18}$$

where $F = \{f(x) = w_{q(x)}\}$ and $q : R^m \to T, w \in R^T$ that q represents the structure of each tree, T represents the number of leaves and each f_t corresponds to an independent tree structure q with the leaf weights being denoted as w. In the LightGBM technique, trees of different structure q are combined, with the structure of each tree being the number of nodes that are created. The loss function that is minimized at any time t is given by the formula:

$$L^{(t)} = \sum_{i=1}^{n} l\left(y_i, \widehat{y}_i^t\right) + \sum_{k=1}^{T} \Omega_{f(t)} \tag{19}$$

The first term measures how well the model adapts to the training data (small values indicate good adaptation) and the second term measures the complexity of each tree, where a new term is introduced in addition to the number of leaves (T), something that results in a reduction in the weights of leaves:

$$\Omega_{f(t)} = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{20}$$

The parameter γ indicates the penalty value for the growth of the tree, so that large values of γ will lead to small trees and small values of γ will lead to large trees. The parameter λ regulates how well the tree weights will shrink, namely an increase of its value leads to the tree weights' shrinkage. Thus:

$$\widehat{y}_i^{(t)} = \sum_{t=1}^{K} f_t(x_i) = \widehat{y}_i^{(t-1)} + f_t(x_i) \tag{21}$$

Thus, the problem is deciding which $f_t(x_i)$ minimizes the loss function at time t:

$$L^{(t)} = \sum_{i=1}^{n} l\left(y_i, \widehat{y}_i^{(t)}\right) + \sum_{k=1}^{T} \Omega_{f(t)} = \sum_{i=1}^{n} l\left(y_i, \widehat{y}_i^{(t-1)} + f_t(x_i)\right) + \sum_{k=1}^{T} \Omega_{f(t)} \tag{22}$$

From the power series expansion Taylor it follows:

$$f(x + \Delta x) \cong f(x) + f^{'}(x)\Delta x + \frac{1}{2} f^{''}(x)(\Delta x)^2 \tag{23}$$

Thus, the resulting relation is:

$$L^{(t)} \cong \sum_{i=1}^{n} \left[l\left(y_i, \widehat{y}_i^{(t-1)}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega_{f(t)} \tag{24}$$

where $g_i = d_{\widehat{y}_i^{(t-1)}} l(y_i, \widehat{y}_i^{(t-1)})$ and $h_i = d^2_{\widehat{y}_i^{(t-1)}} l(y_i, \widehat{y}_i^{(t-1)})$.

Subtracting the constants, the loss function becomes:

$$L^{'(t)} \cong \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega_{f(t)} \tag{25}$$

Putting $I_j = \{i | g(x_i) = j\}$ the set of observations on sheet j, the above relation is reformulated as follows:

$$L^{'(t)} \cong \sum_{i=1}^{n} \left[g_i w_q(x_i) + \frac{1}{2} h_i w_q^2(x_i) \right] + \Omega_{f(t)} = \sum_{i=1}^{T} \left[\left(\sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i + \lambda \right) w_j^2 \right] + \gamma T \tag{26}$$

Setting $G_j = \sum_{i \in I_j} g_i$ and $H_j = \sum_{i \in I_j} h_i$ the following relation emerges:

$$L^{'(t)} = \sum_{i=1}^{T} \left[G_j w_j + \frac{1}{2} \left(H_j + \lambda \right) w_j^2 \right] + \gamma T \tag{27}$$

Assuming that the structure of the tree $(q(x))$ is known, the optimal weight on each leaf is obtained by minimizing the above relation with respect to w_j, so that:

$$w_j = -\frac{G_j}{H_j + \lambda} \tag{28}$$

Subsequently, by replacing w_j, the following equation results, which also calculates the quality of the structure of the new tree:

$$L^{'(t)} = -\frac{1}{2} \sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T \tag{29}$$

Finally, the algorithm creates divisions using the following function:

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda}+\frac{G_R^2}{H_R+\lambda}-\frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right]-\gamma \tag{30}$$

where the first fraction is the score of the left part of the separation, the second fraction is the score of the right part of the separation, the third fraction is the score in case that the separation does not take place and γ measures the cost of the complexity of the separation.

The process of solving a problem begins with creating a tree and growing it up to a specific user-defined depth. The tree is pruned in the divisions with a negative Gain and, then, a truncated version of the new tree is added to the model. The procedure is repeated for K times (K: parameter defined by the user). It is important to note that the LighGBM algorithm, which is characterized by its efficiency, accuracy and speed, creates histograms and uses the generated classes instead of the entire range of each variable's values, achieving a significant reduction in training time. It also grows vertically, which means that it grows at the level of leaf (leaf-wise method, Fig. 3), while other algorithms grow at depth (depth-wise method (Fig. 4)), choosing to grow the leaf with the maximum difference of the cost function. During the leaf-wise tree growth, the algorithm becomes very efficient, as it can significantly reduce the losses, thus gaining accuracy, while at the same time the regression processes are completed quickly.

Another important feature that makes LightGBM one of the most complete and widespread algorithms in Machine Learning is that it does not use all the training data, but a sample of them, which results from the Gradient One Side Sampling method (GOSS). The basic idea of the GOSS methodology focuses on the fact that not all observations contribute the same to the training of the algorithm, since those with a small cost function's first derivative are better trained than those with a large one. Ignoring the observations with a small derivative result in the creation of biased samples and in a definite change in the distribution of data, something which leads to a separation that is greater than the optimal one and to an obvious over-adaptation of the model to the sample. To address the problem, random observations with a small cost function's derivative are selected, which are sorted according to the absolute value of their derivative. Finally, the $a \times 100\%$ with the largest derivative and the $b \times 100\%$ from the rest are selected. For the calculation of the loss function the observations with a small derivative are multiplied by $\frac{1-a}{b}$, thus giving more importance to the poorly trained, without significantly differentiating the distribution of the data. By training only one sample in each iteration, a significant increase in the process of the algorithm learning is achieved, resulting in its fast convergence to the optimal solution. Specifically, for a training set of T with n cases such that $T = \{x_1, x_2, \ldots, x_n\}$, where each x_i is a vector with dimension s in the space X_s. In each iteration of the gradient boosting algorithm, the negative slopes of the cost function in relation to the output of the model are denoted as $G = \{g_1, g_2, \ldots, g_n\}$. Implementing the GOSS method, the cases are classified according to the absolute values of their degrees in descending order. Thus, a set A with the $a \times 100\%$ larger slopes, a set A^c consisting of $(1-\alpha) \times 100\%$ cases with the smallest slopes and a subset B with size $b \times |A^c|$ are created. Then, all cases are classified according to the estimated variance cost in vector $V_j(d)$ on the set $A \subset B$, so that:

$$\overline{V}_j(d) = \frac{1}{n}\left(\frac{\left(\sum_{x_i \in A_l} g_i + \frac{1-a}{b}\sum_{x_i \in B_l} g_i\right)^2}{n_l^j(d)} + \frac{\left(\sum_{x_i \in A_r} g_i + \frac{1-a}{b}\sum_{x_i \in B_r} g_i\right)^2}{n_r^j(d)}\right) \tag{31}$$

where $A_l = \{x_i \subset A : x_{ij} \subset d\}$, $A_r = \{x_i \subset A : x_{ij} \subset d\}$, $B_l = \{x_i \subset B : x_{ij} \subset d\}$ and $B_r = \{x_i \subset B : x_{ij} \subset d\}$, while the coefficient $\frac{1-a}{b}$ is used to normalize the sum of the slopes above B with respect to the magnitude of A^c.

Different statistical strategies for the distribution and management of datasets, also known as validation techniques, may be used to evaluate ML models objectively, both for self-evaluation and for comparison with the corresponding alternative models. In the present study it was used the most frequent and reliable cross-validation approach, the k-fold, in which the dataset is randomly divided into k sections of nearly equal population. One of the k-above subsets is employed as a test subset, while the all-theoretic compound of the remaining k-1 subsets is used as a training subset. A total of k computation cycles are done, with each k subset being used as a test subset in turn. The benefit of this evaluation method is that each dataset is only used once for training and once for testing. The parameter k can attain any positive integer value, although the most common choice in practical applications is k = 5, known as 5-Fold Cross Validation. Fig. 5 depicts the 5-Fold Cross Validation procedure.

The performance metrics of the LighGBM algorithm for the three datasets considered herein were given in Tables B1, B2, and B3

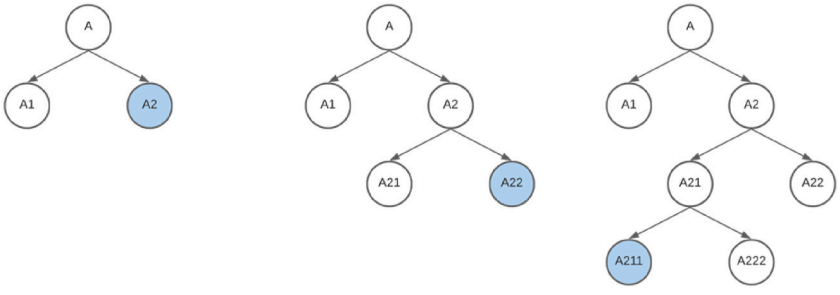

Fig. 3. Leaf-wise method.

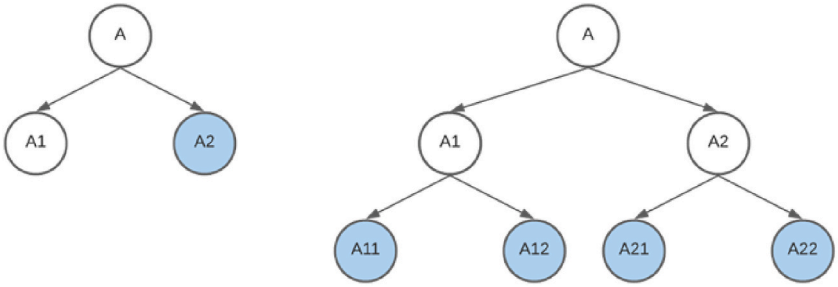

Fig. 4. Depth-wise method.

(Appendix B). Generally, the LighGBM algorithm achieves the highest coefficient of determination, while the error fluctuation remains very low in comparison to the other methods. This gives a clear explanation that a large percentage of data points (91% in the first dataset, 78% in the second dataset, and 89% in the third dataset) fall within the results of the regression equation, therefore the method adapts optimally to the data. Note that in the above Tables some of the most valid error metrics are compared, since, in the forecasting procedure by ML methods, the error measurement between the estimated value and the actual value is useful both to assess the performance of the model and to define the objective function of the model. In any case, the LightGBM approach produces the lowest error, which is explained as high overall performance, training stability, and generalization ability. Finally, the algorithm has satisfactory training times, which can be further improved if the training data are pre-sorted.

Diagrams of the methodology, that show its superiority and the way the LightGBM algorithm works, as well as the way of modeling the problem, are presented in the following. The plots of algorithm for the dataset of the bare buildings are presented in the following Figs. 6–9:

The plots of LightGBM algorithm for the dataset of the infilled buildings are presented in the following Figs. 10–13:

The plots of LightGBM algorithm for the dataset of the buildings with pilots are presented in the following Figs. 14–17:

More specifically, the prediction error plot shows the actual targets from each dataset against the predicted values generated by the model. This allows identifying how much variance exists in the model by comparing them against the 45° line, where the prediction matches exactly the model. Also, the residual plot is a graph that shows the residuals on the vertical axis and the independent variable on the horizontal axis. If the points in a residual plot are randomly dispersed around the horizontal axis, a linear regression model is appropriate for the data; otherwise, a nonlinear model is more appropriate. Moreover, a learning curve is a plot that shows time or experience on the x-axis and learning or improvement on the y-axis. The model is evaluated on the training dataset after each update during training and depicts the measured performance. Finally, the validation curve is a graphical technique that can be used to measure the influence of a single hyperparameter. By looking at this curve, it can be determined if the model is underfitting, overfitting or just-right for some range of hyperparameter values.

6. Additional validation and interpretability of machine learning model

The proposed approach employs specialized tools to assess the method and explain why the model reached a particular decision to make the preceding procedure and findings more understandable. Specifically, it uses an additional validation process to make the produced complete-trust model and extensive experiments in order to be the final model interpretable. By combining these two methods, a completely transparent model can be created, which will explain the hidden correlations that may appear between model

Fig. 5. 5-Fold cross validation.

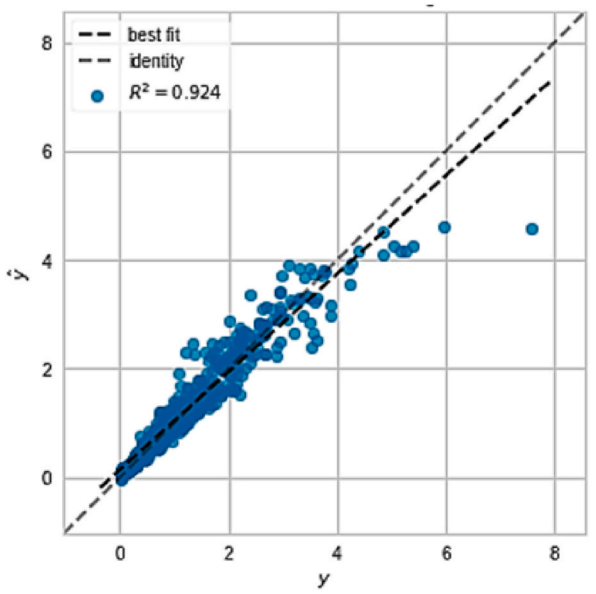

Fig. 6. Prediction Error for LightGBM for 30 iterations.

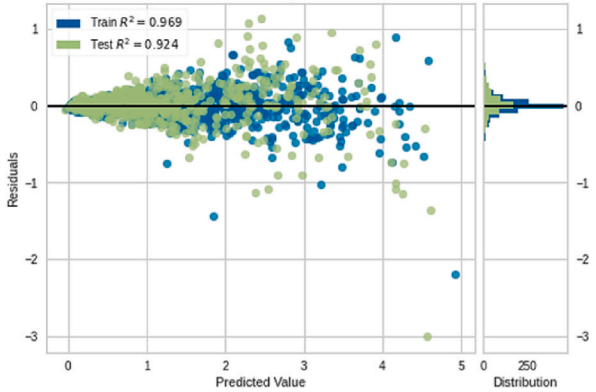

Fig. 7. Residuals for LightGBM for 30 iterations.

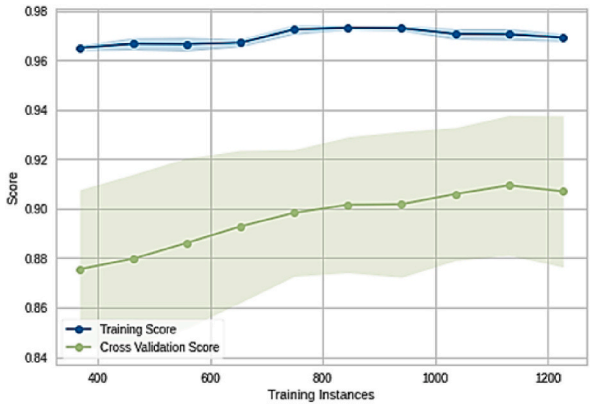

Fig. 8. Learning curve for LightGBM.

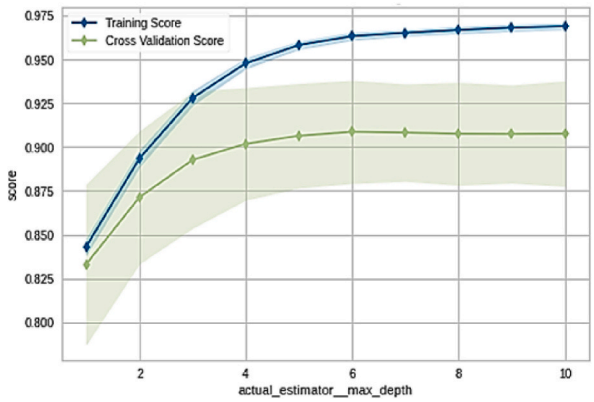

Fig. 9. Validation curve for LightGBM.

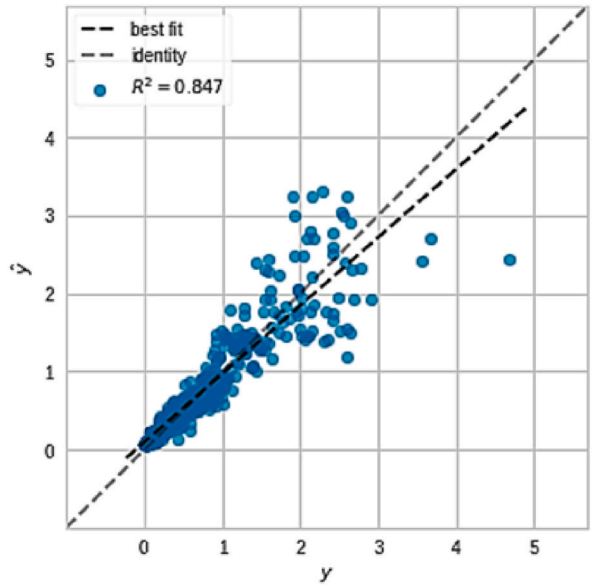

Fig. 10. Prediction Error for LightGBM for 30 iterations.

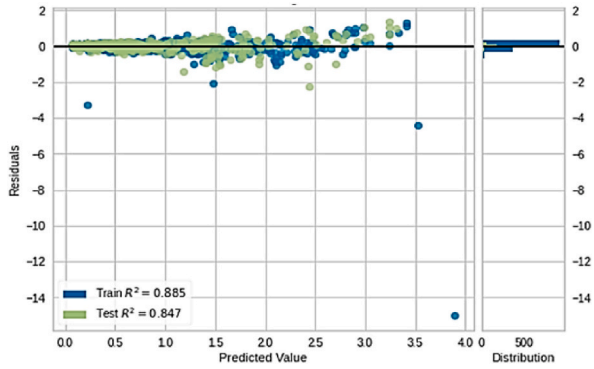

Fig. 11. Residuals for LightGBM for 30 iterations.

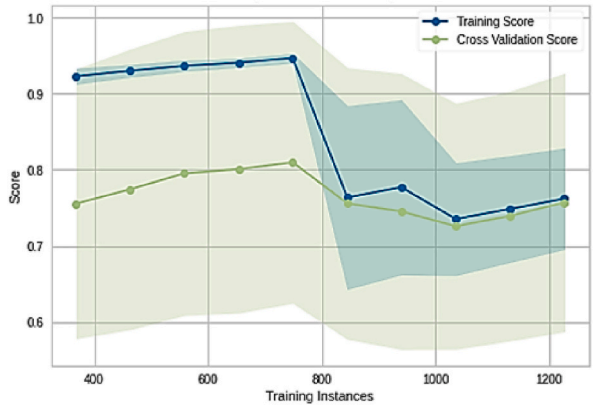

Fig. 12. Learning curve for LightGBM.

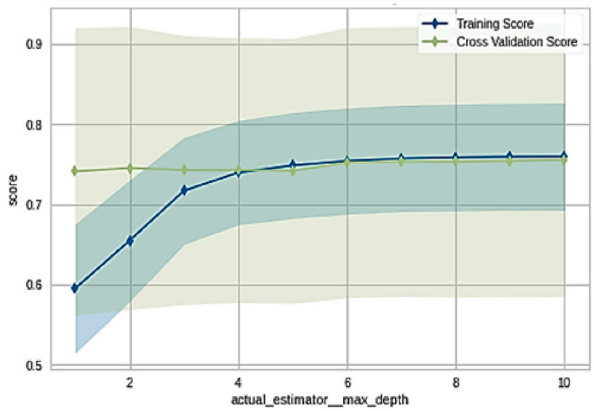

Fig. 13. Validation curve for LightGBM.

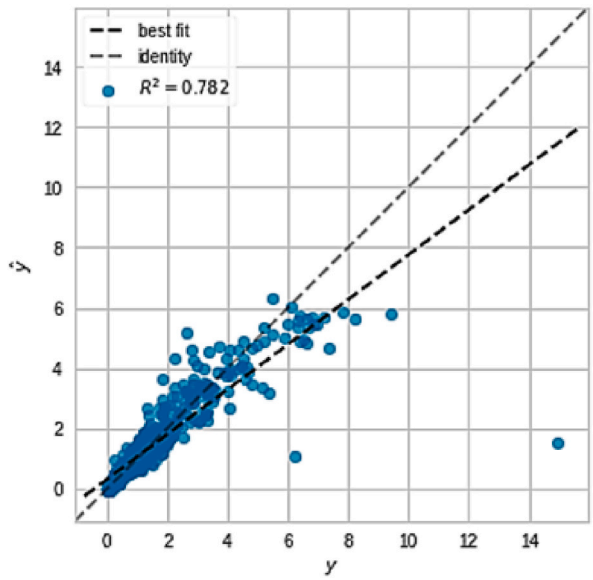

Fig. 14. Prediction Error for LightGBM for 30 iterations.

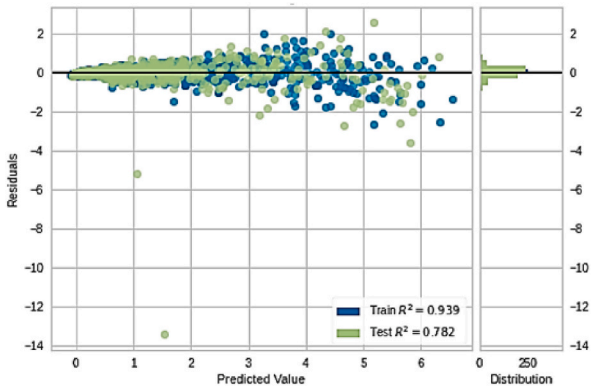

Fig. 15. Residuals for LightGBM for 30 iterations.

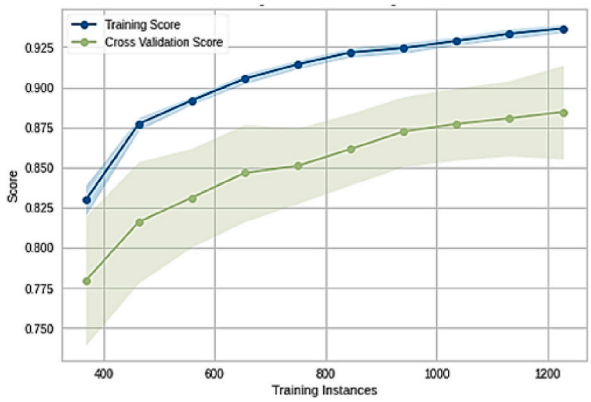

Fig. 16. Learning curve for LightGBM.

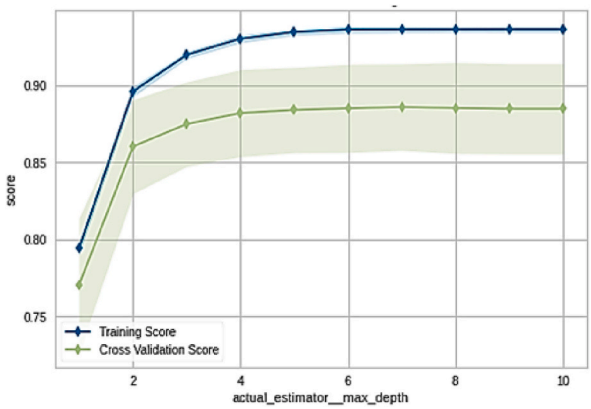

Fig. 17. Validation curve for LightGBM.

parameters and data, the data features with the highest predictability, as well as the influence of each feature on the final prediction in unknown data. So, engineers without technical expertise and experience in machine learning architectures can produce or trust complicated, intelligent data-driven models.

6.1. Additional validation

The extra validation process is critical in directing adequate understanding, mitigation, and evaluation of the model's inherent and foreign hazards. It produces model quality assurance and control, making machine learning development and deployment more interpretable, dependable, and productive. When a model's failure substantially impacts the entire machine learning application, the

importance of model validation cannot be overstated. The proposed extra validation process analyzes the LightGBM model to ensure that it is risk-free (at least to an acceptable degree) before production. It is a model risk assessment exercise that assists in comprehending the worst-case scenario and its impact on associated hazards, including data and idea drifts, low training data quality, malicious attacks, runtime framework defects and issues, model training and evaluation bugs, model explainability issues, fairness assessment concerns and distinguishing production environments from development settings. The suggested approach ensures that models are validated before being sent to production and that their compliance is monitored while in production. A depiction of the classical machine learning pipeline against the machine learning pipeline with additional validation is presented in Fig. 18 below.

The suggested method provides deep extra validations (inspection parameters) by using a series of inspections/checks that deliver reports on insights on data or model configuration errors. The processes are exposed to inspection circumstances that provide prompts such as pass, fail, warning and results while measuring and monitoring performance metrics on the deployment model whilst engaging with real situations.

The first check is about the performance report in unknown data. The following Fig. 19 summarizes the given scores on an unknown dataset [55–57].

As perceived, the train-test scores' relative degradation is not greater than 0.2. Specifically, Neg RMSE: train = −0.17, test = −0.29 and Neg MAE: train = −0.1, test = −0.17. So, reproducibility/robustness, a critical problem in numerical domains such as applied machine learning, was proved in the best way possible. This demonstrates that the acquired findings are not artifacts of a single study lab's unique configuration. Also, the performance score segmented by 2 top (HI and V_{w1}) features in a heatmap, presented in Fig. 20.

The next check is the regression systematic error using the bias ratio. The bias ratio was developed as a result of research into the behavior of data values as they address projected expectations with the valuation of data that determines their performance. It is a tangible statistic that identifies valuation bias or purposeful data manipulation of the regression process by a comparable data point without needing the actual choice to be disclosed (transparent). This metric detects deviations in return distributions that suggest the existence of bias in subjective predictions. A systematic error threshold in model predictions is shown by a non-zero mean of the error distribution. In the present approach, the bias ratio is 2.05E-11, which means that the error distribution indicates the non-presence of systematic error in proposed model predictions. The regression systematic error is presented in Fig. 21 [58–61].

Also, Fig. 22 shows the check regression error distribution.

The kurtosis value was found equal to 23.71598. In probability theory and statistics, kurtosis describes the shape of a probability distribution. Higher kurtosis corresponds to the greater extremity of deviations (or outliers) and not to the data configuration near the mean. The largest over and under estimation of the unknown dataset instances are presented in Tables 3 and 4.

To estimate the unknown parameters in the regression function, it is required to understand how the data at each location in the explanatory variable space relate to the associated regression function value. For example, if the measurement system is used to track how the response variable's values change over time, the deterministic variation in the data would equal the sum of the drift function and the actual regression function. Consequently, to achieve the regression function, either the data must be altered before fitting the

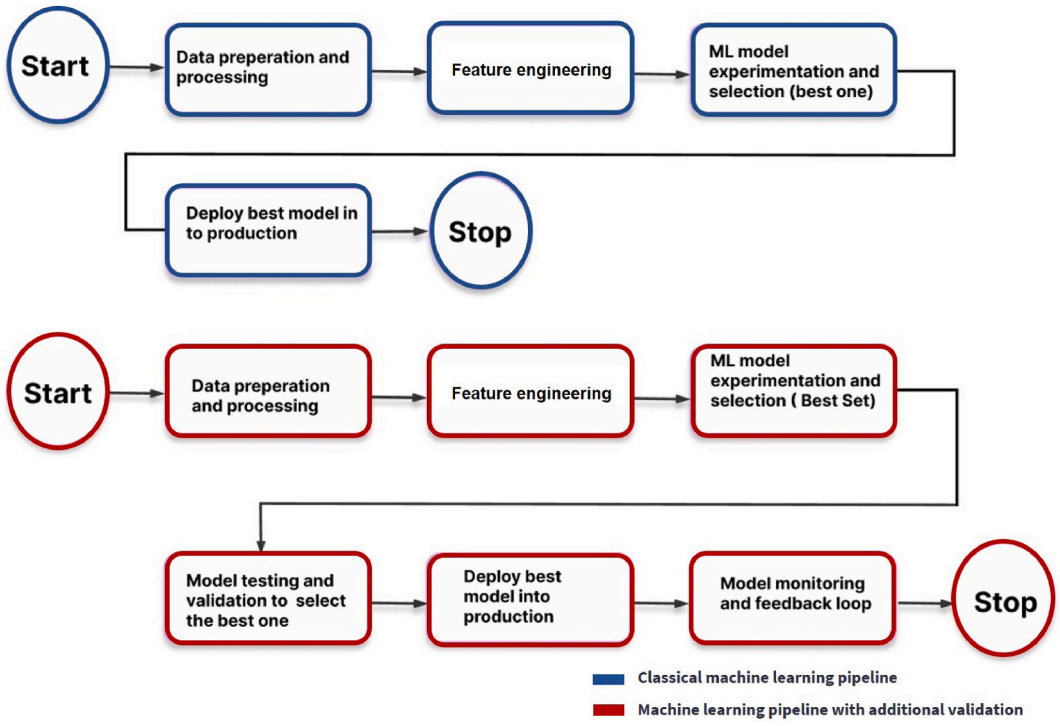

Fig. 18. Classical machine learning pipeline vs machine learning pipeline with additional validation.

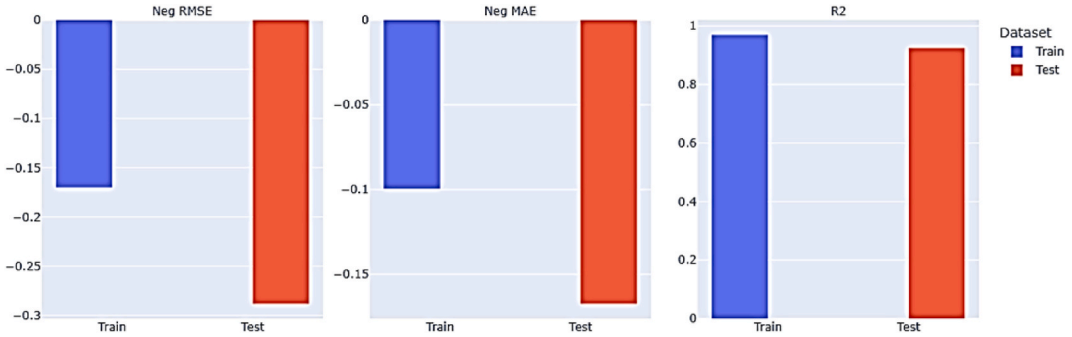

Fig. 19. Performance report.

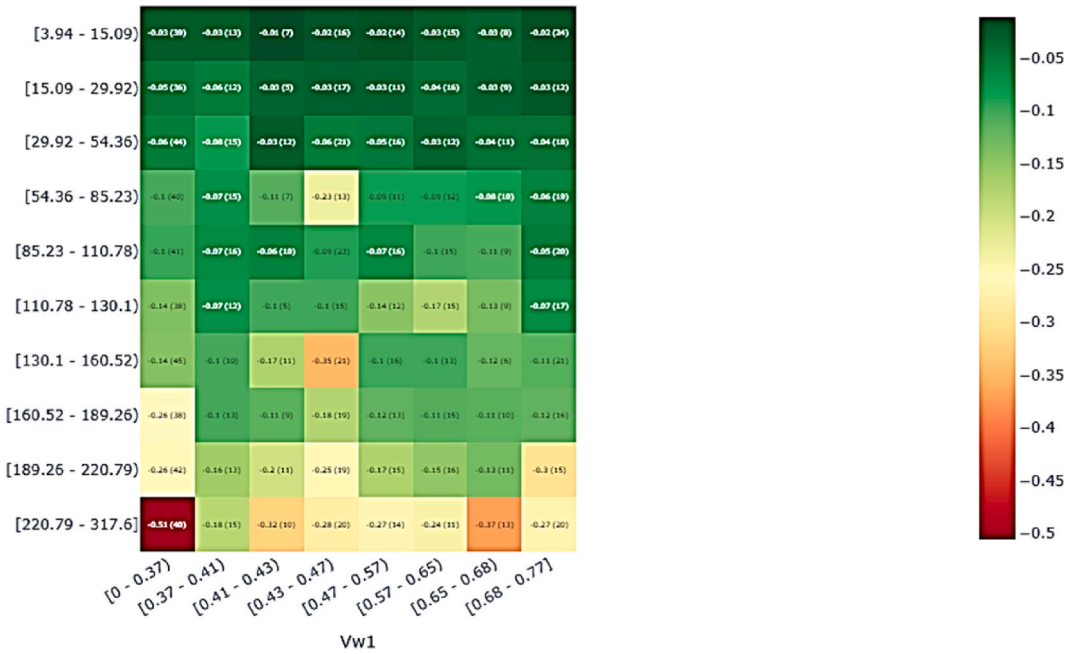

Fig. 20. Segment Performance on unknown data.

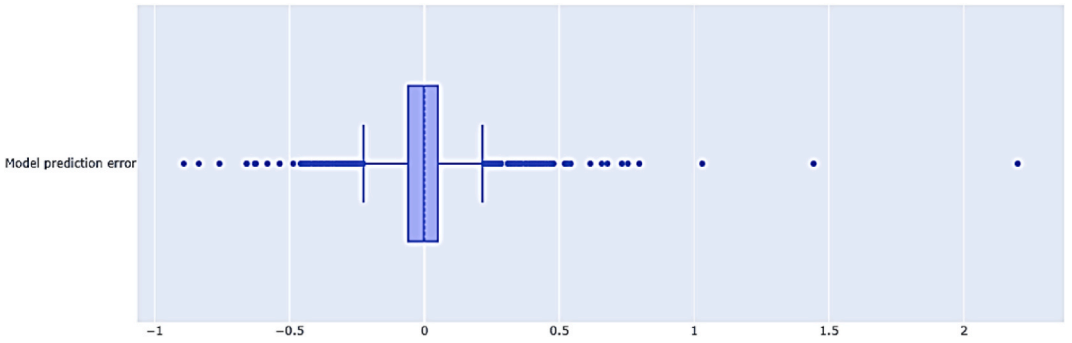

Fig. 21. Regression Systematic Error check.

model or the fitted model must be adjusted after the fact. In either situation, information concerning the drift function's shape would be required. So, the next check is to calculate the prediction drift between train dataset and the test dataset, using statistical measures.

As shown in Fig. 23, the categorical drift score≤0.15 and numerical drift score≤0.075. The model prediction Earth Mover's Distance drift score was found to be 0.02. The drift score is a measure of the difference between two distributions, in this check - the test

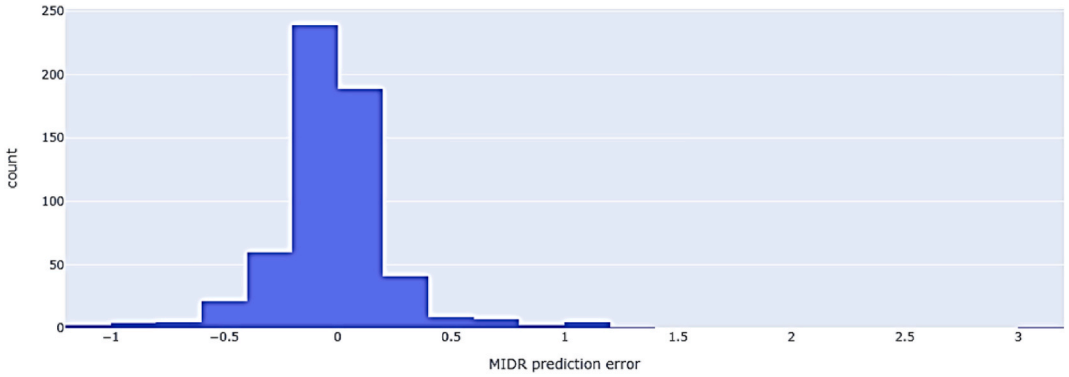

Fig. 22. Histogram of prediction regression error distribution.

Table 3
Largest over estimation errors.

Instance	MIDR	Predicted MIDR	MIDR Prediction Difference
21	7.12	4.92	2.20
240	3.29	1.85	1.44
1878	4.25	3.22	1.03

Table 4
Largest under estimation errors.

Instance	MIDR	Predicted MIDR	MIDR Prediction Difference
335	3.27	4.16	−0.89
216	1.96	2.80	−0.84
1195	2.07	2.83	−0.76

and train distributions [62–64].

Since it would be difficult to generalize an activity like drift correction to a generic process, and since it would also be unnecessary for many processes, most process modeling methods rely on having data in which the observed responses are directly equal, on average, to the regression function values. Another way of expressing this idea is to say the mean of the random errors at each combination of explanatory variable values is zero.

The next check is about overfitting caused by using too many iterations in a gradient boosted model. The check restricts the boosting model to employing up to N estimators per run and displaying the Neg RMSE obtained for each subgroup of estimators for both the training and test datasets. The process's threshold is when the test score does not deviate from the best result across iterations by more than 5%.

The proposed model performs accurately against unseen data, achieving its purpose, as depicted in Fig. 24. The generalization of the model to new, unseen data ultimately allows to use it to make predictions. In addition, a check of the Predictive Power Score of all features was made to estimate each feature ability to forecast the correct value. A depiction of the process is presented in Fig. 25.

The PPS is used to measure a feature's capacity to predict the label independently. In the graph above, we should be suspicious of data errors if:

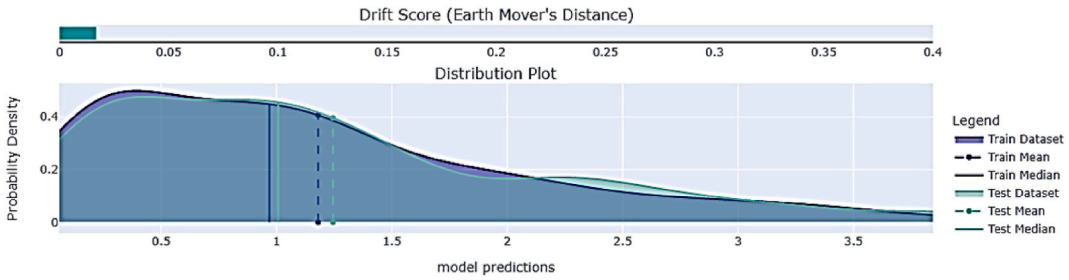

Fig. 23. Prediction Drift check.

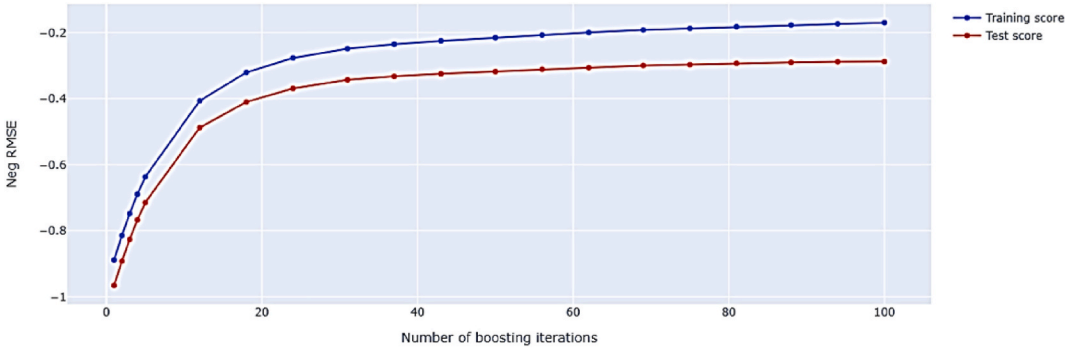

Fig. 24. Boosting Overfit check.

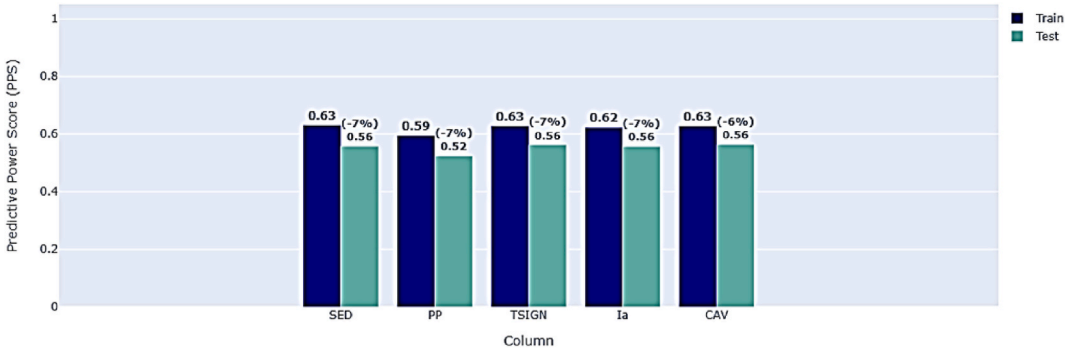

Fig. 25. Feature label correlation change by predictive power score.

1. The PPS values in the training dataset are high: This can indicate that this feature's effectiveness in predicting the label is due to data leaking, implying that the feature formerly held information based on the label.
2. The difference in train and test PPS (train PPS is higher) is significant: An even more strong signal of data leakage may be explained by leaking in the train that is irrelevant to a new dataset, as a feature that was powerful in the train but not in the test.
3. The difference between test and train PPS (test PPS is greater) is significant: An abnormal value might suggest drift in the test dataset, resulting in a coincidental association to the target label.

Finally, an Outlier Probability Score was computed to calculate the local divergence of a particular sample's density concerning its neighbors. The following Table 5 presents the results of the process.

These outlier scores are directly interpretable as a probability of an object being an outlier.

6.2. *Model Explainer [55–69]*

On the other hand, when machine learning models are employed in ways that affect people's lives, it is crucial not only to generalize but also to understand what impacts model behavior in cases such as model debugging (Why did my model make this mistake? How can I develop my model?), human-AI cooperation (How can I understand and trust the model's decisions?), and regulatory compliance (Does my model meet legal requirements?). The interpretability component adds to the "diagnose" step of the model lifecycle process by creating human-understandable descriptions of a Machine Learning model's predictions. It gives different perspectives on a model's behavior: global explanations (for example, what attributes impact the overall behavior of a decision) and local explanations (e.g., why a datapoint was predicted or not). Model explanations may also be shown for a particular cohort as a subset of data points. This is useful for examining the fairness of model predictions for choices in a specific data group.

Model predictions for many instances may be explained using either global model interpretation techniques (on a modular level) or

Table 5
Outlier sample detection.

Instance	Outlier Probability Score	MIDR
1280	0.85	1.25
80	0.79	1.99
1900	0.78	1.68
600	0.78	2.17

individual instance explanations. The global methods may be used on a collection of instances by treating them as the whole dataset and applying the global approaches to this subset. Individual explanation techniques may be applied to each instance before listing or aggregating the results for the entire group. To identify the global and local interpretability of the discussed model, Contributions Plots were used that show the marginal effect that each feature has on the predicted result of the model. In particular, a typical example of the process is shown in Fig. 26 below.

Table 6 shows each individual feature's contribution to the prediction for a specific observation. The contributions (starting from the population average) add to the final prediction. This allows to explain exactly how each individual prediction has been built up from all the individual ingredients in the model. From the table we can see that HI has the maximum contribution among the seismic and structural parameters, something which can be explained by the fact that HI captures the energy induced by the seismic motion to the structural systems. This observation has been also shown in other research studies, for example [19]. Considering the parameters that have the least contribution (TSIGN, TUD, TBRAC), it was found that they are based on the earthquake's duration. This finding is in accordance with previous research studies [19]. However, no certain pattern was found, as it depends on the special characteristics of the individual problem (structures, seismic motions, ML methods).

Also, the Partial Dependency Plot depicts the relationship between the target response and a subset of input characteristics of relevance, while marginalizing the values of all other input features (the complement features).

Fig. 27 depicts how the model prediction changes when one attribute changes. The figure represents a sample of observations and how they might alter with this feature (gridlines). The average impact is shown in grey. The effect of modifying a characteristic for a single Index is shown in blue. The number of observations sampled for the average, the number of gridlines shown, and the number of points along the x-axis for which model predictions are calculated (gridpoints) clearly explain how the model can be changed in a particular situation.

The following plot (Figs. 28 and 29) shows the relation between feature values and Shapley's values. Shapley values are a very effective way of generating explanations from cooperative game theory. The payoff/gain of the players of a cooperative game is given by a real function that gives values to sets of players.

These plots allow investigating the general relationship between feature value and impact on the prediction and relation between feature value and Shap interaction value. Generally speaking, the Shap values is a highly accurate method that can check whether the model uses features in line with intuitions or about the relationships that have been learned between the input features and the predicted outcome.

7. Conclusions

In the present paper an extensive comparative evaluation of a large number of Machine Learning algorithms for the reliable prediction of 3D R/C buildings' seismic response was carried out. In order to accomplish this aim, a large training dataset consisting of 30 R/C buildings with different structural parameters (the number of stories, the structural eccentricity and the ratio of base shear received by R/C walls (if they exist) along the two orthogonal horizontal axes) was selected. The buildings were designed on the basis of provisions of EC8 and EC2. For each one of these buildings three different configurations regarding their masonry infill walls were

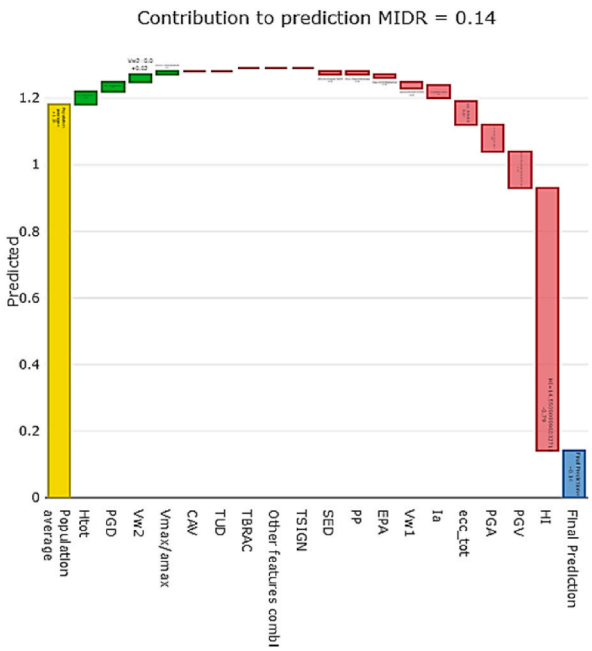

Fig. 26. Contributions plot.

Table 6
Contributions table.

ID	Reason	Effect	ID	Reason	Effect
1	HI = 14.550909996032715	−0.79	10	EPA = 0.0538799986243248	−0.01
2	PGV = 3.620490074157715	−0.11	11	Vmax/amax = 0.06503999978303909	+0.01
3	PGA = 0.05674000084400177	−0.08	12	PP = 0.17664000391960144	−0.01
4	ecc_tot = 0.0	−0.07	13	SED = 30.67694091796875	−0.01
5	Ia = 0.07670000195503235	−0.04	14	TSIGN = 22.64423942565918	−0.0
6	Htot = 9.600000381469727	+0.04	15	CAV = 312.87738037109375	+0.0
7	PGD = 2.2482900619506836	+0.03	15	TUD = 0.05477000027894974	+0.0
8	Vw2 = 0.0	+0.02	17	TBRAC = 2.1169300079345703	+0.0
9	Vw1 = 0.4300000071525574	−0.02	18	Other features combined	+0.0
Average of population 1.18					
Final prediction 0.14					

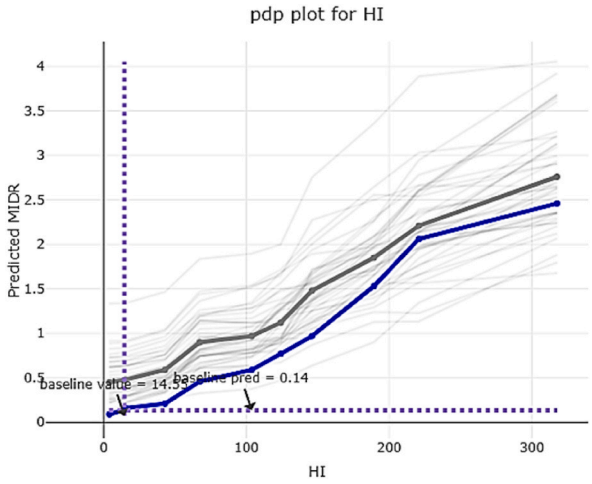

Fig. 27. Partial Dependency Plot for feature HI.

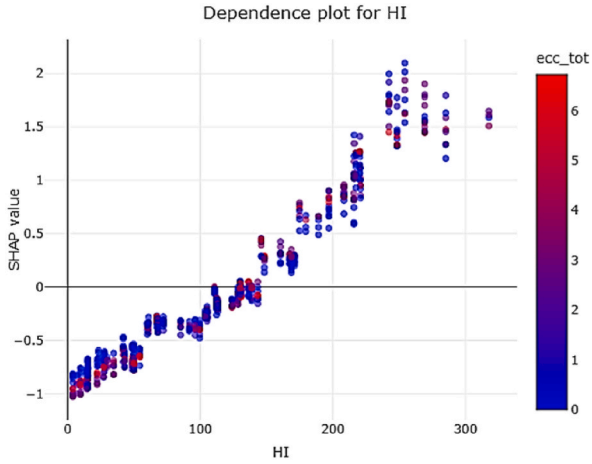

Fig. 28. Shap Dependence (relationship between feature value and Shap value).

assumed (without masonry infills, with masonry infills in all stories and with masonry infills in all stories except for the ground story), leading to three different data subsets consisting of 30 buildings each. The selected buildings were analyzed for 65 appropriately chosen real earthquake records using Nonlinear Time History Analyses. As inputs in the process of Machine Learning methods both seismic and structural parameters widely used in the literature were chosen. The well-documented Maximum Interstory Drift Ratio was selected as the damage index for the R/C buildings. Based on the research study's results, the following conclusions can be drawn:

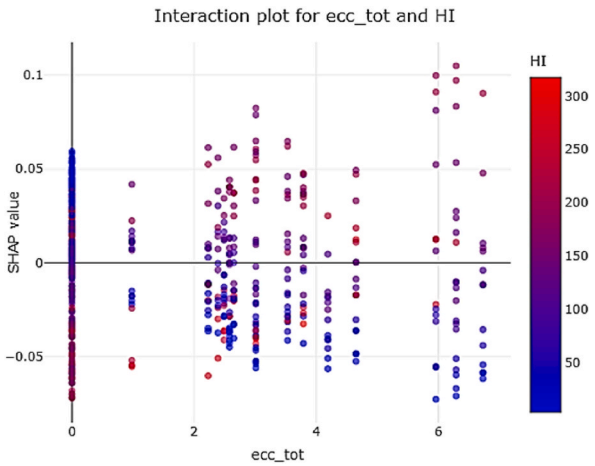

Fig. 29. Interaction Dependence (relation between feature value and shap interaction value).

- Historical data can be utilized in order to develop a realistic model, capable to effectively simulate the earthquake response and to predict with great accuracy the seismic damage of structures belonging to different types.
- The general methodology of the proposed procedure uses the most technologically advanced methods in the field of civil engineering and expands them significantly, as it extracts the hidden knowledge found in structural and seismic data in order to add intelligence to the methods of seismic response prediction, as well as to the mechanisms for optimal decision-making related to seismic risk.
- The high generalizability of the LightGBM algorithm, as well as the convergence stability of the proposed methodology, proves that it is capable of performing well even when the problem is multiparametric.
- The GOSS technique used by the LightGBM algorithm handles with great precision the noisy scattered points of incorrect classification, something those other methodologies cannot handle.
- The tree segmentation method utilized by the algorithm leads to results characterized by remarkable prediction, while offering generalization, which is one of the key requirements in the field of machine learning. Moreover, it reduces bias and variance, as well as eliminates overfitting, implementing a robust forecast model.
- The proposed method, as a problem of multiple spatial-temporal variables, argues that machine learning methods can be utilized in order to solve dynamic problems of high complexity with affordable computational costs.
- The suggested methodology uses specialized tools to analyze the process and explain the model's choice n order to make the procedure and results more robust, transparent, and explainable.
- The proposed procedure constitutes a very promising methodology, which can significantly improve the safety of structures and infrastructure in general under earthquake excitations.

The suggested approach is simple to implement during the research phase and may be described as a collection of offline methods to improve data validation, identify possible methodological difficulties and validate and analyze scenarios. The most important task for the evolution of the proposed methodology is, initially, the process of finding optimization solutions to achieve higher accuracy results. Also, of great importance is the detection of the optimal hyperparameters of the algorithm, in order to enhance the predictive process. Moreover, the training dataset can be expanded to buildings with different structural characteristics and to earthquake records with seismic features of greater range. Finally, the expansion of the methodology with data transformation techniques should be considered, so that the algorithm can locate the optimal representations of the input variables in order to make it easier to extract the useful information. We strongly believe that the proposed methodology can update the civil protection mechanisms to include scientific methods and appropriate technical or modeling tools in their technological systems to make substantial innovative leaps in the new era.

Author statement

Konstantinos Demertzis: Methodology, Software, Analyses, Writing, Lazaros Iliadis: Supervision, Reviewing and Editing, Konstantinos Kostinakis: Methodology, Software, Analyses, Writing, Konstantinos Morfidis: Methodology, Software, Analyses, Writing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

Appendix. A

Table A1
Data for the 65 selected seismic excitations

No	Earthquake name	Date	Magnitude (Ms)	Distance to fault (km)	Component (deg)	PGA (g)
1	Imperial Valley	15/10/1979	6.9	23.8	225/315	0.128/0.078
2	Imperial Valley	15/10/1979	6.9	28.7	012/282	0.27/0.254
3	Kocaeli, (Turkey)	17/8/1999	7.8	144.6	090/180	0.06/0.049
4	Landers	28/6/1992	7.4	128.3	000/270	0.057/0.046
5	Loma Prieta	18/10/1989	7.1	28.2	090/180	0.247/0.215
6	Whittier Narrows	1/10/1987	5.7	25.2	000/090	0.221/0.124
7	Northridge	17/1/1994	6.7	25.4	177/267	0.357/0.206
8	Northridge	17/1/1994	6.7	30	020/110	0.474/0.439
9	N. Palm Springs	8/7/1986	6	43.3	270/360	0.144/0.132
10	Northridge	17/1/1994	6.7	13	000/270	0.41/0.482
11	Northridge	17/1/1994	6.7	6.4	090/360	0.604/0.843
12	Northridge	17/1/1994	6.7	12.3	000/090	0.303/0.443
13	Whittier Narrows	1/10/1987	5.7	10.8	048/318	0.426/0.443
14	Cape Mendocino	25/4/1992	7.1	9.5	000/090	0.59/0.662
15	Chi-Chi (Taiwan)	20/9/1999	7.6	2.94	N/W	0.251/0.202
16	Chi-Chi (Taiwan)	20/9/1999	7.6	10.04	N/W	0.393/0.742
17	Chi-Chi (Taiwan)	20/9/1999	7.6	4.01	N/W	0.162/0.134
18	Chi-Chi (Taiwan)	20/9/1999	7.6	7.31	N/W	0.821/0.653
19	Chi-Chi (Taiwan)	20/9/1999	7.6	11.14	N/W	0.44/0.353
20	Chi-Chi (Taiwan)	20/9/1999	7.6	10.33	N/W	0.13/0.147
21	Chi-Chi (Taiwan)	20/9/1999	7.6	5.92	N/W	0.188/0.148
22	Erzincan (Turkey)	13/3/1992		2.0	NS/EW	0.515/0.496
23	Loma Prieta	18/10/1989	7.1	12.7	000/090	0.367/0.322
24	Loma Prieta	18/10/1989	7.1	14.4	000/090	0.555/0.367
25	Loma Prieta	18/10/1989	7.1	14.5	000/090	0.529/0.443
26	Northridge	17/1/1994	6.7	7.1	090/360	0.583/0.59
27	Northridge	17/1/1994	6.7	8.9	270/360	0.753/0.939
28	Northridge	17/1/1994	6.7	14.6	000/090	0.877/0.64
29	Northridge	17/1/1994	6.7	6.2	052/142	0.612/0.897
30	Campano Lucano (Italy)	23/11/1380	6.9	39	E-W/N-S	0.047/0.048
31	Spitak (Armenia)	7/12/1988	6.7	20	E-W/N-S	0.183/0.183
32	Izmit (Turkey)	17/8/1999	7.6	29	W-E/S-N	0.129/0.091
33	Duzce (Turkey)	12/11/1999	7.2	18	E-W/N-S	0.8/0.745
34	Duzce (Turkey)	12/11/1999	7.2	113	S-N/E-W	0.022/0.021
35	Duzce (Turkey)	12/11/1999	7.2	98	030/120	0.018/0.016
36	Duzce (Turkey)	12/11/1999	7.2	94	E-W/N-S	0.042/0.041
37	Izmit (Turkey)	17/8/1999	7.6	80	E-W/N-S	0.114/0.11
38	Duzce (Turkey)	6/6/2000	6.1	158	LONG/TRAN	0.004/0.004
39	Strofades (Greece)	18/11/1997	6.6	54	261/351	0.053/0.054
40	Aigion (Greece)	15/6/1995	6.5	138	065/155	0.013/0.013
41	Friuli (Italy)	11/9/1976	5.5	7	E-W/N-S	0.105/0.23
42	Volvi (Greece)	4/7/1978		15	E-W/N-S	0.099/0.115
43	Dinar (Turkey)	1/10/1995	6.4	0	W-E/S-N	0.319/0.273
44	Izmit (Turkey)	17/8/1999	7.6	5	E-W/N-S	0.244/0.296
45	Duzce (Turkey)	12/11/1999	7.2	0	W-E/S-N	0.513/0.377
46	Imperial Valley	15/10/1979	6.9	43.6	262/352	0.238/0.351
47	Loma Prieta	18/10/1989	7.1	16.1	000/090	0.417/0.212
48	Loma Prieta	18/10/1989	7.1	77.4	180/270	0.195/0.244
49	Northridge	17/1/1994	6.7	30.9	155/245	0.465/0.322
50	Northridge	17/1/1994	6.7	36.9	090/180	0.29/0.264
51	Duzce, Turkey	12/11/1999	7.3	17.6	000/090	0.728/0.822.
52	Northridge	17/1/1994	6.7	32.7	090/180	0.103/0.186
53	Imperial Valley	15/10/1979	6.9	54.1	075/345	0.122/0.167
54	Superstition Hills	24/11/1987	6.6	18.2	225/315	0.156/0.116
55	Duzce (Turkey)	12/11/1999	7.3	8.2	180/270	0.348/0.535
56	Imperial Valley	15/10/1979	6.9	7.6	002/092	0.213/0.235
57	Imperial Valley	15/10/1979	6.9	4.2	140/230	0.485/0.36
58	Imperial Valley	15/10/1979	6.9	1	140/230	0.519/0.379
59	Imperial Valley	15/10/1979	6.9	1	140/230	0.41/0.439

(*continued on next page*)

Table A1 (*continued*)

No	Earthquake name	Date	Magnitude (Ms)	Distance to fault (km)	Component (deg)	PGA (g)
60	Livermore	27/1/1980	5.5	3.6	270/360	0.258/0.233
61	Superstition Hills	24/11/1987	6.6	13.9	000/090	0.358/0.258
62	Superstition Hills	24/11/1987	6.6	13.3	090/180	0.172/0.211
63	Morgan Hill	24/4/1984	6.1	12.8	270/360	0.224./0.348
64	Imperial Valley	15/10/1979	6.9	12.6	140/230	0.364/0.38
65	Morgan Hill	24/4/1984	6.1	3.4	150/240	0.156/0.312

Appendix B

Table B1
Performance metrics of the compared algorithms for the bare buildings (dataset ROW_FORM_BARE)

Machine Learning Algorithm	Regression Metric					
	R^2	MAE	MSE	RMSE	MAPE	TT (Sec)
Light Gradient Boosting Machine	0.9076	0.1722	0.0867	0.2902	0.1899	0.082
Gradient Boosting Regressor	0.8968	0.1904	0.0968	0.3068	0.2452	0.205
Random Forest Regressor	0.8883	0.1887	0.1035	0.3184	0.1752	0.823
Extra Trees Regressor	0.8840	0.1884	0.1064	0.3237	0.1706	0.657
k-Nearest Neighbors Regressor	0.8343	0.2406	0.1542	0.3875	0.2377	0.065
Linear Regression	0.8312	0.2757	0.1585	0.3939	0.5849	0.019
Bayesian Ridge	0.8312	0.2757	0.1588	0.3941	0.5835	0.018
Ridge Regression	0.8283	0.2768	0.1622	0.3981	0.5804	0.017
Decision Tree Regressor	0.7897	0.2565	0.1941	0.4378	0.2318	0.024
AdaBoost Regressor	0.7721	0.3527	0.2099	0.4578	0.9696	0.143
Elastic Net	0.7650	0.3100	0.2224	0.4675	0.3449	0.020
Lasso Regression	0.7647	0.3100	0.2227	0.4678	0.3484	0.018
Orthogonal Matching Pursuit	0.7550	0.3202	0.2318	0.4776	0.3477	0.018
Huber Regressor	0.7378	0.3438	0.2478	0.4919	0.7261	0.065
Least Angle Regression	0.5082	0.4422	0.4716	0.5778	1.6721	0.021

*TT (Sec) = Training Time in seconds.

Table B2
Performance metrics of the compared algorithms for the infilled buildings (dataset ROW_FORM_FULL-MASONRY)

Machine Learning Algorithm	Regression Metric					
	R^2	MAE	MSE	RMSE	MAPE	TT (Sec)
Light Gradient Boosting Machine	0.7833	0.1535	0.2979	0.3861	0.9794	0.078
Bayesian Ridge	0.7385	0.2045	0.3217	0.4217	1.5333	0.016
Ridge Regression	0.7367	0.2049	0.3230	0.4228	1.4272	0.017
Linear Regression	0.7366	0.2052	0.3240	0.4230	1.4559	0.017
Least Angle Regression	0.7342	0.2082	0.3249	0.4247	1.5213	0.019
k-Nearest Neighbors Regressor	0.6760	0.1549	0.3447	0.4532	0.2621	0.063
Elastic Net	0.6423	0.2640	0.3752	0.4868	2.3327	0.017
Orthogonal Matching Pursuit	0.6378	0.2504	0.3786	0.4880	1.4332	0.016
Decision Tree Regressor	0.6352	0.2614	0.3809	0.4893	1.4516	0.017
Lasso Regression	0.6300	0.2711	0.3814	0.4941	2.2259	0.018
Huber Regressor	0.5998	0.2856	0.4016	0.5117	2.4230	0.060
Gradient Boosting Regressor	0.4942	0.1759	0.4125	0.5323	0.5681	0.182
Random Forest Regressor	0.3898	0.1538	0.4435	0.5410	0.2347	0.764
Extra Trees Regressor	0.2019	0.1567	0.5402	0.5937	0.2384	0.633
AdaBoost Regressor	0.0737	0.3369	0.6196	0.6696	4.8885	0.117

*TT (Sec) = Training Time in seconds.

Table B3
Performance metrics of the compared algorithms for the buildings with pilotis (dataset ROW_FORM_PILOTIS)

Machine Learning Algorithm	Regression Metric					
	R^2	MAE	MSE	RMSE	MAPE	TT (Sec)
Light Gradient Boosting Machine	0.8943	0.2450	0.1999	0.4410	0.2660	0.080
Extra Trees Regressor	0.8812	0.2634	0.2251	0.4698	0.2304	0.660
Random Forest Regressor	0.8792	0.2688	0.2274	0.4726	0.2295	0.808
Gradient Boosting Regressor	0.8717	0.2884	0.2405	0.4854	0.3973	0.197
Decision Tree Regressor	0.8012	0.3373	0.3768	0.6093	0.2777	0.027
Linear Regression	0.7607	0.4667	0.4580	0.6704	1.6584	0.021

(*continued on next page*)

Table B3 (*continued*)

Machine Learning Algorithm	Regression Metric					
	R^2	MAE	MSE	RMSE	MAPE	TT (Sec)
Ridge Regression	0.7603	0.4653	0.4604	0.6717	1.6361	0.015
Bayesian Ridge	0.7602	0.4655	0.4603	0.6717	1.6418	0.019
Least Angle Regression	0.7423	0.4920	0.4876	0.6925	1.6390	0.021
AdaBoost Regressor	0.7411	0.5538	0.4823	0.6919	2.5560	0.145
k-Nearest Neighbors Regressor	0.7350	0.3926	0.5102	0.7037	0.4030	0.063
Huber Regressor	0.6656	0.5199	0.6555	0.7974	1.3578	0.063
Elastic Net	0.6638	0.5223	0.6512	0.7975	1.0582	0.016
Lasso Regression	0.6520	0.5394	0.6736	0.8112	1.1689	0.017
Orthogonal Matching Pursuit	0.6402	0.5574	0.6961	0.8255	1.2555	0.017

*TT (Sec) = Training Time in seconds.

References

[1] ASCE/SEI 41-13, Seismic Evaluation and Retrofit of Existing Buildings, American Society of Civil Engineers (ASCE), Reston, VA, 2014.
[2] ATC, Earthquake Damage Evaluation Data for California, Applied Technology Council, Redwood City, CA, 1985. ATC-13 Report.
[3] FEMA-154, Rapid Visual Screening of Buildings for Potential Seismic Hazards: a Handbook (FEMA-154), second ed., 2002. Washington, DC.
[4] H.-H. Tsang, Ray K.L. Su, T.K. Lam Nelson, S.H. Lo, Rapid assessment of seismic demand in existing building structures, Struct. Des. Tall Special Build. 18 (4) (2009) 427–439, https://doi.org/10.1002/tal.444.
[5] A.J. Kappos, G. Panagopoulos, Ch. Panagiotopoulos, Gr. Penelis, A hybrid method for the vulnerability assessment of R/C and URM buildings, Bull. Earthq. Eng. 4 (2006) 391–413, https://doi.org/10.1007/s10518-006-9023-0.
[6] T. Anagnos, Ch. Rojahn, A. Kiremidjian, NCEER-ATC Joint Study on Fragility of Buildings, Technical Report NCEER 95-0003, National Center for Earthquake Engineering Research, Stateiew York at Buffalo, 1995.
[7] M. Khatibinia, M.J. Fadaee, J. Salajegheh, E. Salajegheh, Seismic reliability assessment of RC structures including soil–structure interaction using wavelet weighted least squares support vector machine, Reliab. Eng. Syst. Saf. 110 (2013) 22–23, https://doi.org/10.1016/j.ress.2012.09.006.
[8] P. Kalakonas, V. Silva, Seismic vulnerability modelling of building portfolios using artificial neural networks, Earthq. Eng. Struct. Dynam. 51 (2) (2021) 310–327, https://doi.org/10.1002/eqe.3567.
[9] F. Soleimani, X. Liu, Artificial neural network application in predicting probabilistic seismic demands of bridge components, Earthq. Eng. Struct. Dynam. 51 (3) (2022) 612–629, https://doi.org/10.1002/eqe.3582.
[10] W. Chen, L. Zhang, An automated machine learning approach for earthquake casualty rate and economic loss prediction, Reliab. Eng. Syst. Saf. 225 (2022), 108645, https://doi.org/10.1016/j.ress.2022.108645.
[11] R. Ghiasi, P. Torkzadeh, M. Noori, A machine-learning approach for structural damage detection using least square support vector machine based on a new combinational kernel function, Struct. Health Monit. 15 (3) (2016) 302–316, https://doi.org/10.1177/1475921716639587.
[12] S.S. Afshari, F. Enayatollahi, X. Xu, X. Liang, Machine learning-based methods in structural reliability analysis: a review, Reliab. Eng. Syst. Saf. 219 (2022), 108223, https://doi.org/10.1016/j.ress.2021.108223.
[13] E. Harirchian, S.E.A. Hosseini, K. Jadhav, V. Kumari, S. Rasulzade, E. Işık, M. Wasif, T. Lahmer, A review on application of soft computing techniques for the rapid visual safety evaluation and damage classification of existing buildings, J. Build. Eng. 43 (2021), 102536, https://doi.org/10.1016/j.jobe.2021.102536.
[14] Y. Xie, M.E. Sichani, J.E. Padgett, R. DesRoches, The promise of implementing machine learning in earthquake engineering: a state-of-the-art review, Earthq. Spectra 36 (4) (2020) 1769–1801, https://doi.org/10.1177/8755293020919419.
[15] H. Sun, H.V. Burton, H. Huang, Machine learning applications for building structural design and performance assessment: state - of - the - art review, J. Build. Eng. 33 (2021), 101816, https://doi.org/10.1016/j.jobe.2020.101816.
[16] O.R. de Lautour, P. Omenzetter, Prediction of seismic-induced structural damage using artificial neural networks, Eng. Struct. 31 (2009) 600–606, https://doi.org/10.1016/j.engstruct.2008.11.010.
[17] M.H. Arslan, An evaluation of effective design parameters on earthquake performance of RC buildings using neural networks, Eng. Struct. 32 (7) (2010) 1888–1898, https://doi.org/10.1016/j.engstruct.2010.03.010.
[18] A. Kia, S. Sensoy, Assessment the effective ground motion parameters on seismic performance of R/C buildings using artificial neural network, Indian J. Sci. Technol. 7 (12) (2014) 2076–2082, https://doi.org/10.17485/ijst/2014/v7i12.26.
[19] K. Morfidis, K. Kostinakis, Seismic parameters' combinations for the optimum prediction of the damage state of R/C buildings using neural networks, Adv. Eng. Software 106 (2017) 1–16, https://doi.org/10.1016/j.advengsoft.2017.01.001.
[20] K. Morfidis, K. Kostinakis, Approaches to the rapid seismic damage prediction of r/c buildings using artificial neural networks, Eng. Struct. 165 (2018) 120–141, https://doi.org/10.1016/j.engstruct.2018.03.028.
[21] K. Morfidis, K. Kostinakis, Comparative evaluation of MFP and RBF neural networks' ability for instant estimation of r/c buildings' seismic damage level, Eng. Struct. 197 (2019), 109436, https://doi.org/10.1016/j.engstruct.2019.109436.
[22] Y. Zhang, H.V. Burton, H. Sun, M. Shokrabadi, A machine learning framework for assessing post-earthquake structural safety, Struct. Saf. 72 (2018) 1–16, https://doi.org/10.1016/j.strusafe.2017.12.001.
[23] Y. Zhang, H.V. Burton, Pattern recognition approach to assess the residual structural capacity of damaged tall buildings, Struct. Saf. 78 (2019) 12–22, https://doi.org/10.1016/j.strusafe.2018.12.004.
[24] EN1998-1 (Eurocode 8, Design of Structures for Earthquake Resistance - Part 1: General Rules, Seismic Actions and Rules for Buildings, European Committee for Standardization, 2005.
[25] EN1992-1-1, Eurocode 2, Design of Concrete Structures, Part 1-1: General Rules and Rules for Buildings, European Committee for Standardization, 2005.
[26] TOL-Engineering Software House, RAF Version 3.3: Structural Analysis and Design Software, Iraklion, Crete, Greece, 2012.
[27] S. Otani, Inelastic analysis of RC frame structures, J Struct Div ASCE 100 (7) (1974) 1433–1449.
[28] Imbsen Software Systems, XTRACT Version 3.0.5: Cross-Sectional Structural Analysis of Components, 2006. Sacramento, CA, http://www.trcbridgedesignsoftware.com/software-XTRACT.html.
[29] F.J. Crisafulli, A.J. Carr, R. Park, Analytical modelling of infilled frames structures - a general review, Bull NZ Soc Earthq Eng 33 (1) (2000) 30–47, https://doi.org/10.5459/bnzsee.33.1.30-47.
[30] N. Tarque, L. Candido, G. Camata, E. Spacone, Masonry infilled frame structures: state-of-the-art review of numerical modelling, Earthquakes and Structures 8 (1) (2015) 225–251, https://doi.org/10.12989/eas.2015.8.1.225.
[31] F.J. Crisafulli, Seismic Behaviour of Reinforced Concrete Structures with Masonry Infills, Ph.D. Thesis, University of Canterbury, Christchurch, New Zealand, 1997, https://doi.org/10.26021/1979.

[32] EN1996-1-1 (Eurocode 6, Design of Masonry Structures - Part 1-1: General Rules for Reinforced and Unreinforced Masonry Structures, European Committee for Standardization, 2005.
[33] S.L. Kramer, Geotechnical Earthquake Engineering, Prentice-Hall, 1996.
[34] SeismoSoft. SeismoSignal v.5.1.0, 2014. www.seismosoft.com.
[35] A.J. Carr, Ruaumoko – A Program for Inelastic Time-History Analysis: Program Manual, Department of Civil Engineering, University of Canterbury, New Zealand, 2006. https://www.researchgate.net/publication/277249247_Ruaumoko_3D_Manual.
[36] PEER (Pacific Earthquake Engineering Research Centre), Strong Motion Database, 2003. https://ngawest2.berkeley.edu/.
[37] European Strong-Motion Database, 2003. http://isesd.hi.is/ESD_Local/frameset.htm.
[38] F. Naeim, The Seismic Design Handbook, second ed., Kluwer Academic, Boston, 2011.
[39] S.K.V. Gunturi, H.C. Shah, Building specific damage estimation, in: Proceedings of 10th World Conference on Earthquake Engineering, Balkema, Madrid. Rotterdam, 1992, pp. 6001–6006.
[40] G. Ke, Meng Qi, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, LightGBM: a highly efficient gradient boosting decision tree, in: Advances in Neural Information Processing Systems, 31st Conference on Neural Information Processing Systems (NIPS 2017), 2017. Long Beach, CA, USA.
[41] J. Elith, J.R. Leathwick, T. Hastie, A working guide to boosted regression trees, J. Anim. Ecol. 77 (2008) 802–813, https://doi.org/10.1111/j.1365-2656.2008.01390.x.
[42] L. Breiman, Random forests, Mach. Learn. 45 (2001) 5–32, https://doi.org/10.1023/A:1010933404324.
[43] P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees, Mach. Learn. 63 (2006) 3–42, https://doi.org/10.1007/s10994-006-6226-1.
[44] D. Bremner, E. Demaine, J. Erickson, J. Iacono, S. Langerman, P. Morin, G. Toussaint, Output-sensitive algorithms for computing nearest-neighbor decision boundaries, Discrete Comput. Geom. 33 (2005) 593–604, https://doi.org/10.1007/s00454-004-1152-0.
[45] N.R. Draper, H. Smith, Applied Regression Analysis, third ed., John Wiley, 1998, ISBN 978-0-471-17082-2.
[46] B.P. Carlin, T.A. Louis, Bayesian Methods for Data Analysis, third ed., Taylor and Francis Group, 2008 https://doi.org/10.1201/b14884.
[47] D. Conniffe, J. Stone, A critical view of ridge regression, Journal of the Royal Statistical Society Series D (The Statistician) 22 (3) (1973) 181–187, https://doi.org/10.2307/2986767.
[48] B. Kamiński, M. Jakubczyk, P. Szufel, A framework for sensitivity analysis of decision trees, Cent. Eur. J. Oper. Res. 26 (1) (2017) 135–159, https://doi.org/10.1007/s10100-017-0479-6.
[49] Kégl Balázs, The return of AdaBoost.MH: multi-class Hamming trees. https://doi.org/10.48550/arXiv.1312.6086, 2013.
[50] H. Zou, T. Hastie, Regularization and variable selection via the elastic Net, J. Roy. Stat. Soc. 67 (2) (2005) 301–320, https://doi.org/10.1111/j.1467-9868.2005.00503.x.
[51] Y. Jiang, Y. He, H. Zhang, Variable selection with prior information for generalized linear models via the prior LASSO method, J. Am. Stat. Assoc. 111 (513) (2016) 355–376, https://doi.org/10.1080/01621459.2015.1008363.
[52] F. Mendels, P. Vandergheynst, J.-P. Thiran, Matching pursuit-based shape representation and recognition using scale-space, Int. J. Imag. Syst. Technol. 16 (5) (2006) 162–180, https://doi.org/10.1002/ima.20078.
[53] T. Hastie, R. Tibshirani, J. Friedman, The elements of statistical learning, in: Springer Series in Statistics, second ed., Springer, Berlin, 2008.
[54] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least Angle regression, Ann. Stat. 32 (2) (2004) 407–499, https://doi.org/10.1214/009053604000000067.
[55] M. Gong, A novel performance measure for machine learning classification, Int. J. Manag. Inf. Technol. 13 (1) (2021), https://doi.org/10.5121/ijmit.2021.13101. Electronic copy available at: https://ssrn.com/abstract=3807764.
[56] Y. Jiao, P. Du, Performance measures in evaluating machine learning based bioinformatics predictors for classifications, Quant. Biol. 4 (2016) 320–330, https://doi.org/10.1007/s40484-016-0081-2.
[57] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Inf. Process. Manag. 45 (4) (2009) 427–437, https://doi.org/10.1016/j.ipm.2009.03.002.
[58] J. Bien, I. Gaynanova, J. Lederer, C. Müller, Prediction Error Bounds for Linear Regression with the TREX, 2018, 01394, https://doi.org/10.48550/arXiv.1801.01394 arXiv:1801.
[59] S. Lipovetsky, W.M. Conklin, Meaningful regression analysis in adjusted coefficients Shapley value model, Model Assisted Statistics Appl. 5 (4) (2010) 251–264, https://doi.org/10.3233/MAS-2010-0170.
[60] M. Taddy, Distributed multinomial regression, Ann. Appl. Stat. 9 (3) (2015), https://doi.org/10.1214/15-AOAS831.
[61] A. Sakata, Estimator of prediction error based on approximate message passing for penalized linear regression, J. Stat. Mech. Theor. Exp. (2018), 063404, https://doi.org/10.1088/1742-5468/aac910.
[62] A. Liu, G. Zhang, J. Lu, Concept drift detection based on anomaly analysis, in: International Conference on Neural Information Processing, 2014, pp. 263–270, https://doi.org/10.1007/978-3-319-12637-1_33.
[63] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, G. Zhang, Learning under concept drift: a review, IEEE Trans. Knowl. Data Eng. (2018), https://doi.org/10.1109/TKDE.2018.2876857, 1–1.
[64] H. Yu, T. Liu, J. Lu, G. Zhang, Automatic Learning to Detect Concept Drift, 2021, 01419, https://doi.org/10.48550/arXiv.2105.01419 arXiv:2105.
[65] K. Demertzis, L. Iliadis, P. Kikiras, A lipschitz - shapley explainable defense methodology against adversarial attacks, in: Artificial Intelligence Applications and Innovations. AIAI 2021 IFIP WG 12.5, International Workshops, Cham, 2021, pp. 211–227, https://doi.org/10.1007/978-3-030-79157-5_18.
[66] D. Hou, T. Driessen, H. Sun, The Shapley value and the nucleolus of service cost savings games as an application of 1-convexity, IMA J. Appl. Math. 80 (6) (2015) 1799–1807, https://doi.org/10.1093/imamat/hxv017.
[67] X. Bao, X. Li, Cost allocation of integrated supply based on shapley value method, in: International Conference on Intelligent Computation Technology and Automation, INSPEC Accession Number, 2010, 11472243, https://doi.org/10.1109/ICICTA.2010.406, 2010.
[68] B. Guo, S. Hao, G. Cao, H. Gao, Profit distribution of liner alliance based on shapley value, J. Intell. Fuzzy Syst. 41 (4) (2021) 5081–5085, https://doi.org/10.3233/JIFS-189993.
[69] K. Demertzis, K. Tsiknas, D. Takezis, C. Skianis, L. Iliadis, Darknet traffic big-data analysis and network management for real-time automating of the malicious intent detection process by a weight Agnostic neural networks framework, Electronics 10 (7) (2021), https://doi.org/10.3390/electronics10070781.