Contents lists available at ScienceDirect

Engineering Structures

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

Classification of buildings' potential for seismic damage using a machine learning model with auto hyperparameter tuning

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

[a] Department of Civil Engineering, Aristotle University of Thessaloniki, Aristotle University Campus, 54124 Thessaloniki, Greece
[b] Earthquake Planning and Protection Organization (EPPO-ITSAK), Terma Dasylliou, 55535 Thessaloniki, Greece
[c] School of Science and Technology, Informatics Studies, Hellenic Open University, Aristotle 18, 26335 Patra, Greece
[d] School of Engineering, Department of Civil Engineering, Democritus University of Thrace, Kimmeria, Xanthi, Greece

ARTICLE INFO

Keywords:
Machine learning
Hyperparameter tuning
Bayesian optimization
Seismic damage prediction
Structural vulnerability assessment
Seismic risk assessment

ABSTRACT

The research on the application of machine learning (ML) methods in the field of earthquake engineering shows a continuous and rapid progress in the last two decades. ML methods and models belonging to the category of supervised, unsupervised and semi-supervised learning are applied for the assessment of seismic vulnerability of structures and estimation of the expected level of seismic damage. These models lead to the classification of seismic damage into predefined classes through the extraction of patterns from data collected from various sources. However, the lack of detailed knowledge can affect their performance and ultimately reduce their reliability, as well as the generalizability that should characterize them. Towards this direction, the present paper attempts to compare and evaluate for the first time the ability of an extensive number of ML methods in the correct classification of R/C buildings at the first stage pre- and post-seismic inspection considering three seismic damage categories. A database consisting of 5850 training samples is used for this evaluation. This database is generated by solving 90 R/C buildings for 65 actual seismic excitations applying nonlinear time history analyses. For each one of the training samples the maximum interstory drift ratio is calculated as a damage index. In addition, a major contribution of this paper is the presentation and extensive documentation of the procedures required for the preprocessing of the data. Finally, an auto hyperparameter tuning method for the winning algorithm is proposed, so that the hyperparameters are automatically optimized utilizing Bayesian Optimization. The most significant conclusion extracted is that the studied ML algorithms extract very different classification results. In addition, the Support Vector Machine - Gaussian Kernel algorithm extracted the most accurate results of all the studied ML algorithms.

1. Introduction

A large number of existing buildings were constructed in countries with regions of high seismicity. These structures were designed using older seismic codes that did not incorporate the most recent provisions, thus leading to high seismic vulnerability under earthquake excitations. For these buildings it is especially crucial to develop a rapid, but also reliable and efficient, method for classifying the seismic damage potential and for prioritizing the buildings with high seismic vulnerability, so that the authorities will be able to develop appropriate earthquake safety plans for seismic rehabilitation. Since now, several researchers have proposed such procedures, some of which were adopted by seismic code guidelines (e.g., see [1-8]). The most of these codes utilize simplified procedures in order to assess the seismic response and the structural damage level, based on certain input parameters such as structural configuration and seismic motion intensity measures. Additionally to these methods, a number of researchers have developed techniques for the rapid estimation of the buildings' seismic vulnerability based on the application of statistical theory, e.g., seismic fragility curves (e.g. [9-19]). These techniques are characterized by certain shortcomings (small number of input structural and seismic parameters, linear relationship between inputs and outputs, simple formulae for the estimation of the damage level based on the input variables), which make their use rather limited and, in many cases, not effective, as they

* Corresponding author.
E-mail addresses: kkostina@civil.auth.gr (K. Kostinakis), konmorf@gmail.com (K. Morfidis), demertzis.konstantinos@ac.eap.gr (K. Demertzis), liliadis@civil.duth.gr (L. Iliadis).

https://doi.org/10.1016/j.engstruct.2023.116359
Received 10 January 2023; Received in revised form 3 May 2023; Accepted 18 May 2023
Available online 7 June 2023
0141-0296/

are not able to capture the full complexity of the relationship between damage and input parameters.

In order to overcome the abovementioned limitations, in the last decades, modern methods based on the adoption of Machine Learning (ML) algorithms were developed. The up-to-date research on these methods has shown that they can provide a fast, reliable, and computationally easy way for classify the buildings' the seismic damage potential and that they can successfully identify structural performance under seismic motions by extracting patterns from data collected via various sources. Machine learning is one of the most important and widespread fields of artificial intelligence that includes those computational methods of studying and constructing algorithms which can learn from appropriate datasets. Their success is based on the thorough processing of the data that record the behavior of a system, so that by detecting the appropriate patterns valuable information can be extracted. Based on this experience the ML algorithms are able to make accurate future predictions. The concept of experience refers to the hidden knowledge contained in the data collected from the field and related to the type of damage suffered by the buildings under investigation. In recent years it has been proven that ML algorithms have the ability to be successfully applied in many areas of modeling engineering problems, giving a serious breakthrough to modern earthquake engineering. More specifically, several research studies have found that ML methods, mainly Artificial Neural Networks (ANNs), can effectively assess the seismic response of complex structures. A thorough literature review of the most commonly used and recently proposed ML methods for the buildings' seismic damage assessment has been made by Harirchian et. al [20], by Xie et al [21] and Sun et al. [22]. Next, a brief review of some of the most significant relevant researches oriented to earthquake engineering is given. Rafiq et al [23] adopted several different types of ANNs (Multi-layer Perceptron, Radial Basis Networks and normalized Radial basis Networks) in order to solve engineering problems. Aoki et al. [24] tried to assess the seismic vulnerability of chemical industrial plants with different topologies with the aid of probabilistic ANNs. In another research study, Lautour and Omenzetter [25] investigated 2D reinforced concrete frames that varied in topology, stiffness, strength and damping and were subjected to a suite of ground motions. They established the ability of the ANNs to reliably estimate the earthquake-induced damage level of these structures. Tesfamariam and Liu [26] studied eight different statistical damage classification techniques in order to estimate the reported seismic induced damage and proved the feasibility and effectiveness of the selected statistical approaches to classify the damage of R/C buildings. Similarly, Arslan [27] created a dataset for the training of ANNs by means of incremental static pushover analyses in order to estimate the ANNs' ability to predict the seismic damage level of medium and high-rise R/C buildings. Kia and Sensoy [28] used a combination of ANNs with SVM in order to classify the damage of R/C slab-column frames. Kostinakis and Morfidis carried out a number of research studies [29-34] in order to assess the efficiency of ANNs regarding the instant prediction and classification of R/C buildings' seismic damage. More recently, Zhang et al. [35] used predictive models including classification and regression tree and Random Forests in order to probabilistically identify the structural safety state of an earthquake-damaged building. The same research team, in another research work [36], adopted several ML techniques for the adequate estimation of the residual structural capacity of damaged tall buildings. A different approach was given by Harirchian and Lahmer [37], who developed a novel method based on type-2 fuzzy for earthquake vulnerability assessment of buildings via Rapid Visual Screening. Mangalathu et. al. [38], using data from the 2014 South Napa earthquake, examined the ability of ML methods, such as discriminant analysis, k-nearest neighbors, decision trees, and random forests, to rapidly estimate seismic building damage. Other research teams conducted also a number of works [36,39-41] in an attempt to thoroughly investigate the applicability of a series of ML techniques to predict the potential of structures for earthquake-induced damage. Similar scientific investigation was conducted by Harirchian and his research teams [42-46]. In addition, the continuous research on the application of ML in earthquake engineering leads to a significant number of relative papers published in the last 2 years (see e.g. [47,48,49,50,51]).

The results of the most research works established the capability of ML methods in the successful seismic damage classification of structures. However, there is a rather limited number of researches that used a large number of ML methods, structures and seismic motions in order to comparatively evaluate the ML techniques' efficiency in estimating the seismic damage response with adequate reliability. Data-driven intelligent systems may translate human knowledge and experience into optimally correct and timely judgments. The lack of detailed knowledge and expertise associated with the use of complex machine learning architectures, on the other hand, can affect the performance of the intelligent model, prevent the adjustment of some critical hyperparameters, and ultimately reduce the algorithm's reliability and generalization, which should characterize these systems. These drawbacks restrict stakeholders, especially civil engineers, from trusting machine learning technologies and using them effectively and consistently. To address the challenge mentioned above, this paper proposes a holistic system that automates the selection and application of the most appropriate algorithmic hyperparameters that optimally solve a problem under consideration, approaching it as a model for finding algorithmic solutions where the problem is solved by mapping between input and output data. Specifically, the present paper attempts a comparative evaluation of a large number of Machine Learning algorithms for the reliable classification of R/C buildings' potential for seismic damage. Moreover, an extensive documentation of the procedures required for the preprocessing of the data is given. For the most appropriate algorithm (winner algorithm), an auto fine-tuning technique based on the Bayesian Optimization (BO) method is proposed to identify the optimal hyperparameters which can optimally solve the given problem. The implementation is based on a fully automated intelligent way and does not require human intervention. For this aim, a training dataset consisting of 30 3D R/C buildings with different structural parameters was chosen. The buildings were designed based on the provisions of EN1992-1–1 [52] and EN1998-1 [53]. For each one of these buildings three different configurations as regards their masonry infills were considered (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 with 30 buildings each. Then, the buildings were analysed my means of the Nonlinear Time History Analyses method (NTHA) for 65 appropriately chosen real earthquake records. Both seismic and structural parameters widely used in the literature were selected as inputs in the process of Machine Learning methods. The quantification of the buildings' damage level was done by means of the well-documented Maximum Interstory Drift Ratio (MIDR). The methodology of the proposed assessment/information system uses and extends the most technologically advanced methods of forecasting, analysis, and modeling of seismic engineering, as it extracts the hidden knowledge found in digital data to adds intelligence to the best decision support methods. At the same time, it gives the stimulus for the utilization of intelligent methods and their penetration in the development sector, for giant innovative leaps and development of previously impossible activities.

2. Dataset generation

A large training dataset consisting of buildings with different structural characteristics was used to generate the database for the training and testing of the ML models. The structures have characteristics that are common to buildings designed and built on the basis of modern seismic codes and according to the construction practice in most european countries with regions of high seismicity. In particular, 30 R/C buildings with structural systems consisting of members in two perpendicular directions (axes × and y) were selected. The buildings are rectangular in plan (dimensions L_xxL_y) and regular in elevation and in plan according to the criteria set by EN1998-1 [53]. The structures differ in the total height H_{tot} (H_{tot} = (stories' number) × (stories' height: 3.2 m)), the value of structural eccentricity e_0 (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 × and y): n_{vx} and n_{vy}. A detailed description of the investigated buildings can be found in [31]. The influence of the masonry infill walls, the placement of which along the height of the buildings is part of the traditional building practice, on the structures' seismic response and damage was considered taking into account for each one of the 30 structures three different assumptions about their distribution. More specifically, three different training subsets were generated: (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 (structures with pilotis). Consequently, the total number of buildings studied herein is 30 different structural systems × 3 different distributions of masonry infills = 90. The three subsets of the buildings were trained separately by the same Machine Learning 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 the buildings. The 30 selected bare buildings (no infill walls) were modeled, analyzed and designed based on the provisions of EN1992-1–1 and EN1998-1. After the elastic modeling and design of the bare buildings, the three subsets mentioned above (bare, infilled, buildings with pilotis) were created and their nonlinear behavior was simulated, in order to analyze them by means of NTHA. Moreover, the masonry infills were modeled as single equivalent diagonal struts with stress–strain diagrams according to the model proposed by Crisafulli [54]. A detailed description and documentation of the design and modeling process of the investigated buildings can be found in [31]. A suite of 65 pairs of horizontal bidirectional earthquake records taken from the PEER [55] and the European Strong-Motion database [56] was chosen in such a way as to cover a large variety of conditions regarding tectonic environment, modified Mercalli intensity and closest distance to fault rapture, thus representing a wide range of intensities and frequency content. A detailed description and documentation of the selected earthquake records can be found in [31].

The 90 buildings (three subsets of 30 buildings each) were subjected to each one of the 65 earthquake ground motions, for which NTHA was conducted with the aid of Ruaumoko software [57]. As a consequence, a total of 5850 NTHA (90 buildings × 65 earthquake records) were conducted herein. For each one of the analyses, the estimation of the seismic damage was determined using the Maximum Interstory Drift Ratio (MIDR), which corresponds to the maximum story's drift among the perimeter frames. The MIDR is extensively adopted as a reliable indicator of structural and nonstructural global damage of R/C buildings (e. g. [58,59]) and has been used by many researchers for the assessment of the building' inelastic response. The values of MIDR have been classified by many researchers. Herein, the classification given by Masi et al. [60] (Table 1) has been adopted. Note that the number of the damage classes (three) was also selected in order to be compatible with the commonly used rationale of seismic damage classification in slight (green), moderate (yellow) and heavy (red) damage states which are utilized in case of the rapid seismic assessment of buildings after strong events.

3. Inputs and outputs

For real problem modeling situations such as the one under consideration, the input models come from the same boundary distribution or follow a common cluster structure. Thus, the classified data enable a learning process, providing useful information for exploring the data structure of the overall set and finding patterns capable of identifying the problem, thus creating an intelligent classification framework. The classification concerns the classification of each sample in one of the predefined classes after successful training. The training of a model of machine learning with the method of classification is called the process in which the function $\hat{f} : R^N \rightarrow T$ is calculated, where T is a set of labels denoting the class. In this problem, the basic evaluation criterion was considered to be the error for a wrong prediction, which depends on the concept of the success of including a sample in the correct class.

For the purposes of the study, both structural and seismic parameters were chosen as input features in the process of the ML methods. More specifically, the following structural parameters (see also Table 2), that are considered crucial for the vulnerability assessment of 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 × and y (ratio n_{vx} and ratio n_{vy}) and the structural eccentricity e_0.

Regarding the seismic parameters, the 14 seismic parameters presented in Table 3 were chosen (e.g. [61,62]). Regarding the output feature, the abovementioned MIDR was chosen, as a reliable damage measure that can adequately capture the damage level of the R/C buildings.

4. Preprocessing of data

The preprocessing of the data refers to the preliminary checks and work carried out on the abovementioned dataset before the use of the ML algorithms, in order to determine if the initial data suffer from various types of problems and if so, then to select the appropriate procedure to deal with them. This process is particularly important because, in case that the quality of the data used is not guaranteed, the performance of the ML algorithms will not be satisfactory or will produce biased or untrue results. Finally, it should be noted that there are a number of techniques which can be used in the preprocessing procedures and that the choice of the best strategy depends on the nature of the examined problem and on the corresponding available data used. In detail, the data pre-processing procedures that were applied to the present dataset include the checks [63,64] presented in the sub-sections.

4.1. Missing values

A Missing Values check was performed and it was found that there is

Table 1
Relation between MIDR and damage state.

MIDR (%)	<0.50 Class 0	0.50-1.00 Class 1	>1.00 Class 2
Degree of damage	Slight (No damages or repairable slight damages)	Moderate (Significant but repairable damages)	Heavy (Non-repairable damages)

Table 2
The structural parameters of the R/C buildings used for the generation of the training dataset.

	Name	n_{vx}	n_{vy}	H_{tot} (m)	L_x (m)	L_y (m)	T_x(sec)	T_y(sec)	e_0 (m)
1	SFxy_3	0.00	0.00	9.60	13.50	10.00	0.699	0.715	0.00
2	SFxy_5	0.00	0.00	16.00	20.00	14.00	0.718	0.719	0.00
3	SFxy_7	0.00	0.00	22.40	20.00	14.00	1.117	1.123	0.00
4	SWxy_3	0.73	0.76	9.60	15.00	10.00	0.416	0.459	0.00
5	SWxy_5	0.77	0.80	16.00	19.00	16.40	0.693	0.649	0.00
6	SWxy_7	0.57	0.64	22.40	19.00	16.40	1.034	0.961	0.00
7	SFExy_3	0.41	0.41	9.60	15.00	15.00	0.534	0.534	0.00
8	SFExy_5	0.46	0.50	16.00	21.00	18.50	0.800	0.836	0.00
9	SFExy_7	0.43	0.46	22.40	21.00	18.50	1.178	1.232	0.00
10	SFExFy_3	0.43	0.00	9.60	17.00	12.50	0.513	0.704	0.00
11	SFExFy_5	0.41	0.00	16.00	20.00	15.00	0.791	0.999	0.00
12	SFExFy_7	0.38	0.00	22.40	20.00	15.00	1.135	1.446	0.00
13	SWxFy_3	0.77	0.00	9.60	15.00	10.00	0.386	0.722	0.00
14	SWxFy_5	0.68	0.00	16.00	20.00	15.00	0.666	0.999	0.00
15	SWxFy_7	0.51	0.00	22.40	20.00	15.00	0.999	1.447	0.00
16	AFxy_3	0.00	0.00	9.60	13.00	9.00	0.678	0.787	0.980
17	AFxy_5	0.00	0.00	16.00	17.50	10.00	0.790	0.885	2.575
18	AFxy_7	0.00	0.00	22.40	17.50	10.00	1.134	1.279	2.387
19	AFExy_3	0.52	0.46	9.60	13.50	9.00	0.493	0.513	4.643
20	AFExy_5	0.43	0.42	16.00	16.00	14.50	0.754	0.727	4.192
21	AFExy_7	0.37	0.36	22.40	16.00	14.50	1.130	1.091	3.795
22	AFExFy_3	0.47	0.00	9.60	13.50	9.00	0.515	0.775	2.226
23	AFExFy_5	0.38	0.00	16.00	16.00	14.50	0.757	0.925	2.649
24	AFExFy_7	0.35	0.00	22.40	16.00	14.50	1.092	1.303	2.491
25	AWxFy_3	0.64	0.00	9.60	14.50	9.00	0.320	0.612	3.523
26	AWxFy_5	0.69	0.00	16.00	14.00	16.00	1.014	0.590	3.012
27	AWxFy_7	0.65	0.00	22.40	14.00	16.00	1.512	0.897	3.010
28	AWxy_3	0.64	0.58	9.60	13.50	10.00	0.325	0.353	6.732
29	AWxy_5	0.65	0.72	16.00	16.25	16.25	0.730	0.641	6.284
30	AWxy_7	0.59	0.67	22.40	16.25	16.25	1.061	0.941	5.963

no unavailable information that could mislead the algorithms and produce untrue results.

4.2. Outliers

An extreme value is defined as a point that is very far from the mean value of the corresponding random variable representing a feature. Samples with feature values very different from the mean value produce significant errors, especially if they are the result of noise during the measurement process, something which has disastrous results for the training process. Distance is measured relative to a threshold, which is usually a multiple of the standard deviation. For a random variable following a normal distribution, a distance that equals twice the standard deviation covers 95% of the points and a distance which equals three times the standard deviation covers 99% of the points. If the number of extreme values is small, then either the values remain and are appropriately modified or these samples are simply discarded, which is the most popular tactic.

One of the most popular methods for finding extreme values is the Interquartile Range (IQR) technique. IQR is the difference between the 3rd (Q3) and the 1st (Q1) quadrant, namely IQR = $Q^3 – Q^1$. Quadrants divide the data into 4 equal parts (quarters), with the intra-quadratic range comprising an intermediate 50% of observations. The remaining 50% is outside this range, with the 25% of the observations being smaller than Q^1 and the remaining 25% being larger than Q^3. A depiction of the IQR method is presented in the following Fig. 1.

It should be emphasized that the extreme values in the case of the problem of seismic damage that is considered herein are important and consist the question of the problem, since, based on them, important decision-making mechanisms are activated (e.g., further detailed seismic evaluation of the building etc.), so it was considered appropriate to seek them, but not to isolate - remove them from the datasets. This decision was considered essential in order to create objective training samples, which will be able to generalize and to better respond to new data, as well as to be able to predict corresponding damage for future periods (forecasting).

4.3. Normalization

Normalization is a process of transforming data, in which numeric values are replaced by corresponding ones, but which are in a certain range of values. This process is usually performed in order to address problems related to the operation or performance of the algorithms. For example, some algorithms perform better when the input values lie in the range [0,1], while in case of algorithms that calculate the distances among the observations, the normalization of values is required in order to deal with the problem that the variables with large values are those that mostly determine the distance of the observations, while small-value variables have very little effect on the distance and, consequently, play no role in calculating the result. In the present research study, the normalization was done with the aid of the method Max-Min. According to this method, all numerical values match with values that fluctuate within a predetermined range based on a linear transformation. Considering a variable *A*, with max_A and min_A being the largest and smallest values respectively, we can match all the values with corresponding ones that fluctuate within a range with a lower limit of new_min_A and an upper limit of new_max_A according to the Eq. (1):

$$x^{'} = \frac{x - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A \tag{1}$$

where *x* is the value of the variable *A* and *x´* is the new value.

This method has the advantage that the user predefines the value range, setting new_min_A and new_max_A, while maintaining the ratio between the values that existed in the original data. On the other hand, the normalization of Max-Min is not appropriate in cases where the data contain extreme values, as they gather the vast majority of values in a minimal part of the value range and use the rest of the part for exceptions.

4.4. Feature reduction

In most cases a set of data can contain too many features, which may be related to each other, provide irrelevant information to the specific

Table 3
Examined ground motion parameters.

Ground Motion Parameter	Range of values	Calculation procedure	Category
Peak Ground Acceleration: **PGA**	(0.004 ÷ 0.822) g	max|a(t)|	Seismic parameters determined from the time histories of the records.
Peak Ground Velocity: **PGV**	(0.86 ÷ 99.35) cm/sec	max|v(t)|	
Peak Ground Displacement: **PGD**	(0.36 ÷ 60.19) cm	max|d(t)|	
Arias Intensity: $\mathbf{I_a}$	(≈0.0 ÷ 5.592) m/sec	$I_a = (\pi/2g) \bullet \int_0^{t_{tot}} [a(t)]^2 dt$	
Specific Energy Density: **SED**	(1.24 ÷ 16762.8) cm^2/sec	$SED = \int_0^{t_{tot}} [v(t)]^2 dt$	
Cumulative Absolute Velocity: **CAV**	(14.67 ÷ 2684.1) cm/sec	$CAV = \int_0^{t_{tot}} \|a(t)\| dt$	
Acceleration Spectrum Intensity: **ASI**	(0.003 ÷ 0.633) g·sec	$ASI = \int_{0.1}^{0.5} S_a(\xi = 0.05) dT$	Seismic parameters determined from the response spectra of the records.
Housner Intensity: **HI**	(3.94 ÷ 317.6) cm	$HI = \int_{0.1}^{2.5} PSV(\xi = 0.05, T) dT$	
Effective Peak Acceleration: **EPA**	(0.003 ÷ 0.63) g	$EPA = (1/2.5)\{\overline{S}_a(\xi = 0.05, T)\}_{0.1}^{0.5}$	
V_{max}/A_{max} **(PGV/ PGA)**	(0. 036 ÷ 0.336) sec	max|v(t)|/max|a(t)|	Seismic parameters accounting for the earthquake's frequency content.
Predominant Period: **PP**	(0.077 ÷ 1.26) sec	$PP = T[\max S_a(\xi = 0.05, T)]$	
Time of Uniform Duration: **TUD**	(≈0.0 ÷ 17.68) sec	Special algorithm (e.g. SeismoSoft [62])	Seismic parameters based on the earthquake's duration.
Time of Bracketed Duration: **TBD**	(≈0.0 ÷ 61.87) sec		
Time of Significant Duration: **TSD**	(1.74 ÷ 50.98) sec		

Where, a(t), v(t) and d(t) are the acceleration, velocity and displacement time history respectively, S_a is the acceleration spectrum, PSV is the pseudovelocity spectrum, ξ is the damping ratio, Time of Uniform Duration is the total time during which the ground acceleration is larger than a given threshold value (usually 5% of PGA), Time of Bracketed Duration is the total time elapsed between the first and the last excursions of a specified level of acceleration (usually 5% of PGA), Time of Significant Duration is the interval of time over which a proportion of the total Arias Intensity is accumulated (usually the interval between the 5% and 95% thresholds).

problem or produce noise, something which reduces the efficiency of the algorithm used. Two depictions of the correlation matrix of the dataset used (regression depiction in left and heat map depiction in right) are presented in the following Figs. 2 and 3.

Also, if the vector space of features has many dimensions (i.e., many features), the volume of this space increases very fast, so the data for the problem will be sparse, causing problems for the methods that try to achieve statistical significance. The amount of data needed in order to be considered dense increases exponentially in relation to the dimension of the feature space. This phenomenon is also known as the "curse of dimensionality". It should also be noted that a large number of features increases the number of parameters of the learning system, and, therefore, its complexity, without this meaning that it will have a correspondingly better performance. Because of these observations, the number of features should be kept as small as possible in order to achieve high system performance.

The solution to these problems is provided with the aid of techniques of dimensional reducing, which offer an efficient solution to managing multidimensional data, as they seek for a low-dimensional structure in multidimensional data. These techniques are considered necessary preprocessing procedures in such cases, as the distances between the data in the reduced space are calculated faster, the size of the dataset is reduced, the data structure which remains hidden in the original multidimensional space is revealed and the efficiency of ML algorithms is significantly improved. The most well-known linear dimensional reduction technique is Principal Component Analysis (PCA).

This method tries to calculate the axes in which the maximum data scatter is observed. For example, for the data $\{X_1, X_2, \cdots, X_n,\} \in R^D$, the covariance table $S = X \bullet X^T$ is calculated, then their average value μ is calculated, the eigenvalues I_i and the eigenvectors e_i are calculated through of the process of self-analysis of S, $I_i \bullet e_i = S \bullet e_i$, and, finally, the d largest eigenvectors are selected and based on them the new variables are calculated by the Eq. (2):

$$Y_i = [e_1, e_2, \cdots, e_d]^T \times (X_i - \mu) \tag{2}$$

Subsequently, a PCA test was performed for the dataset considered in the present study, in order to detect data covariance and to apply, if necessary, a feature reduction. As can be seen from the scree plot in Fig. 4, the principal components retain less than 60% of the statistical data from the original data, so no feature reduction is required.

4.5. Feature selection

This is the process of the optimal selection of a subset of existing features without transformation, in order to retain the most important of them reducing this way their number and at the same time retaining as much useful information as possible. This step is crucial because if features with low separating ability are selected, the resulting learning system will not have satisfactory performance, while if features that provide useful information are selected, the system that will be designed

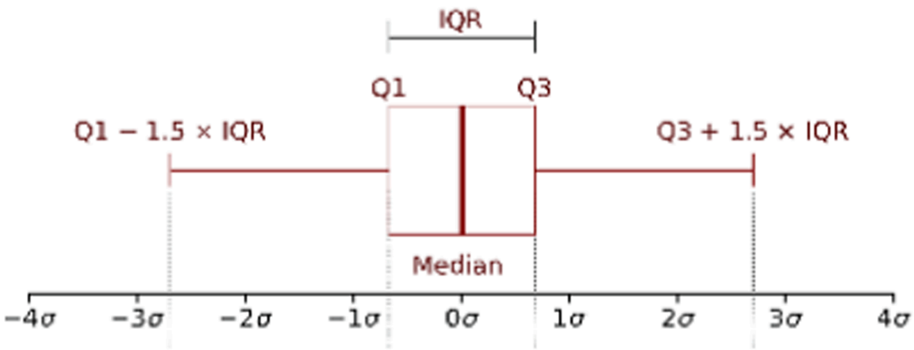

Fig. 1. Interquartile range.

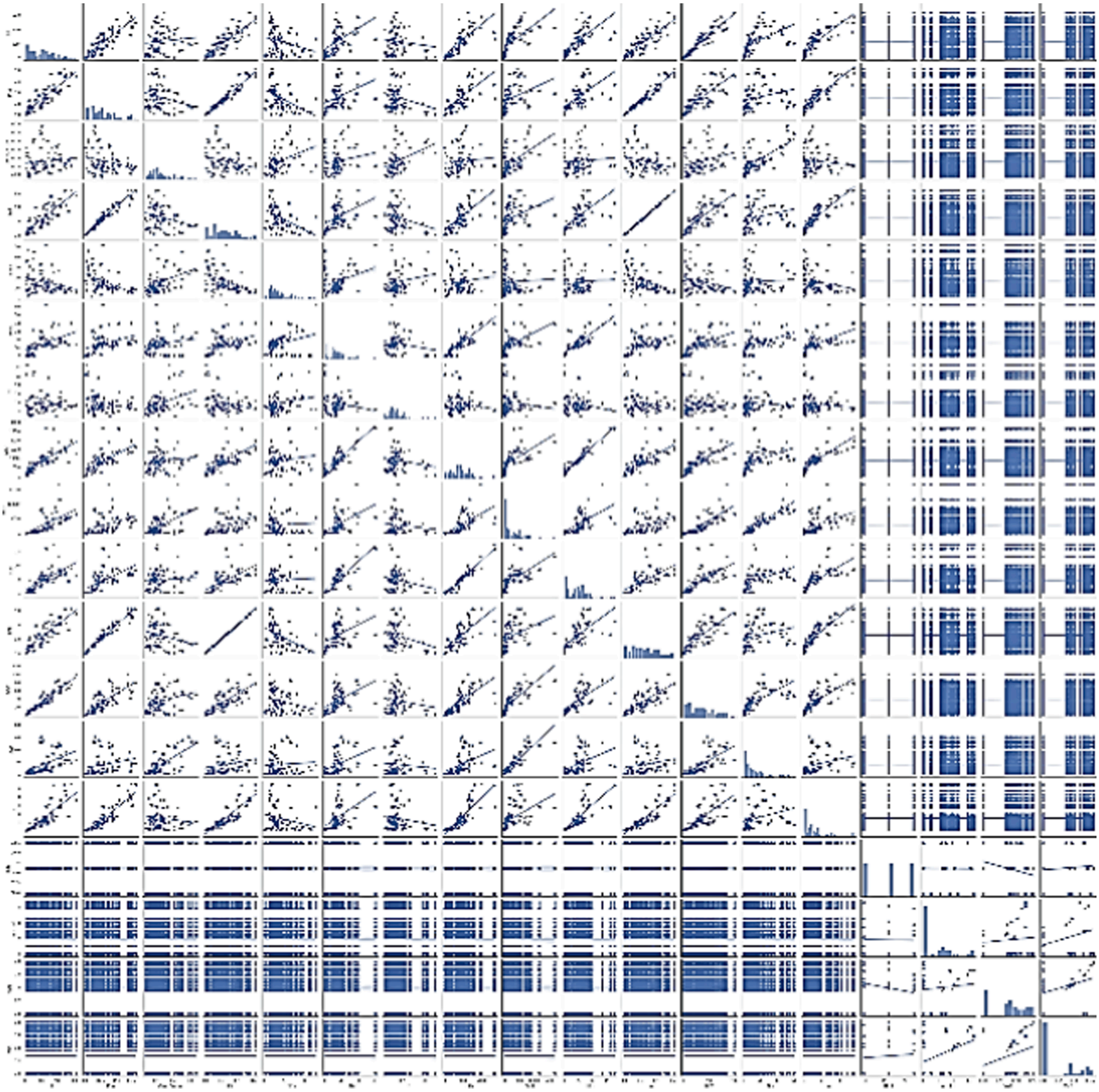

Fig. 2. Correlation matrix (regression depiction).

will be simple and efficient.

One of the strategies that can be followed is to examine the characteristics one by one through a measure of class reparability and to reject those that have a low separating ability. The aim is to select these characteristics that lead to large distances between the groups of samples to and small variation among the same group. This means that the characteristics should get distant values for different classes and close values for the same class, a strategy known as filtering. One of the most popular filtering methods is Forward Selection, which starts with an empty set of selected features. Then, this method selects the most important of the other features, subtracts it from the original set and adds it to the set of selected features. Finally, from the remaining features, the most important one is selected and it is added to the set of selected features. The process is repeated until an output condition is satisfied.

Also, another approach to feature selection is achieved by examining the various combinations of features available and controlling those combinations that lead to higher performance, regardless of the quality of the individual features, an approach which is called wrapping. These methods, in order to select the important features, use the same algorithm that will be applied to the final ML process. In other words, these are not independent methods the results of which can be separately dealt with, but methods that differ in terms of the algorithm and of the solution search technique.

In the present research work, taking into account the inability of classical correlation analysis methods to detect nonlinear correlations such as sinus wave, quadratic curve, etc., the Predictive Power Score (PPS) technique was chosen to summarize the most important features between available features. PPS can work with nonlinear relationships, but also with asymmetric relationships, explaining that variable A in-

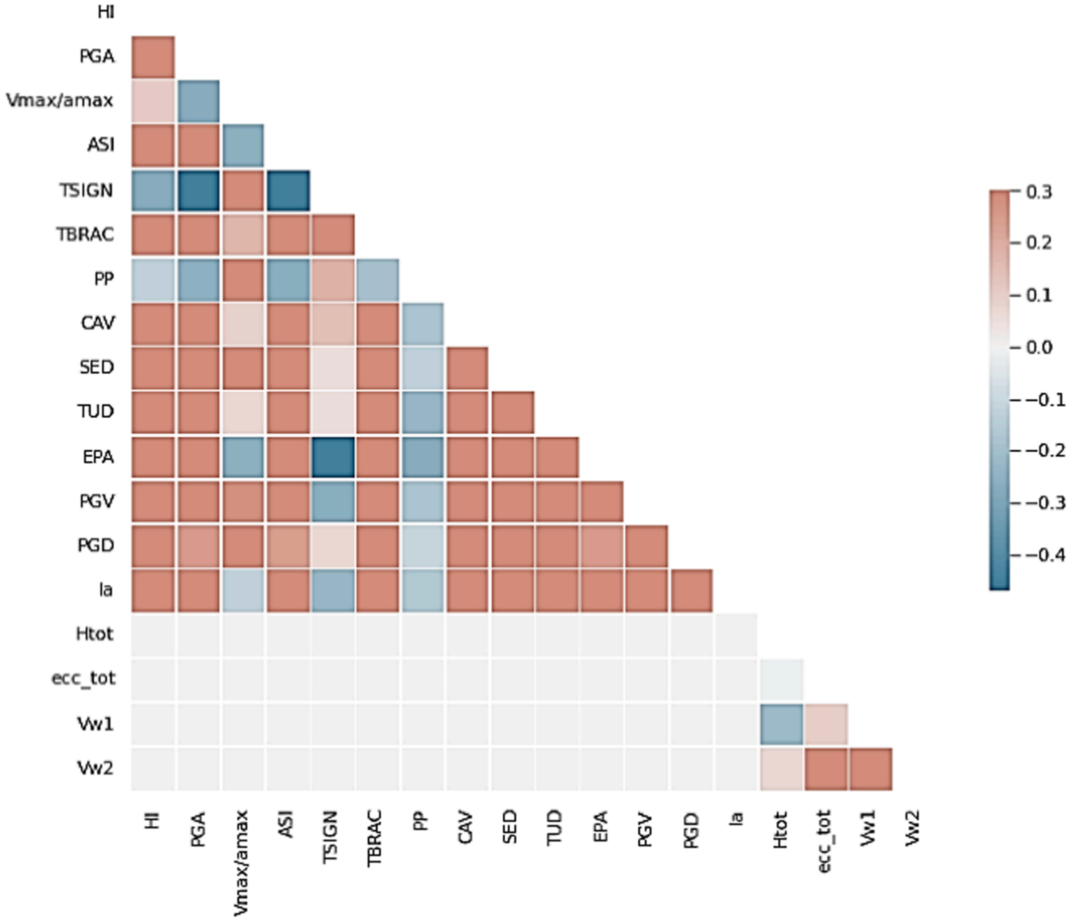

Fig. 3. Correlation matrix (heat map depiction).

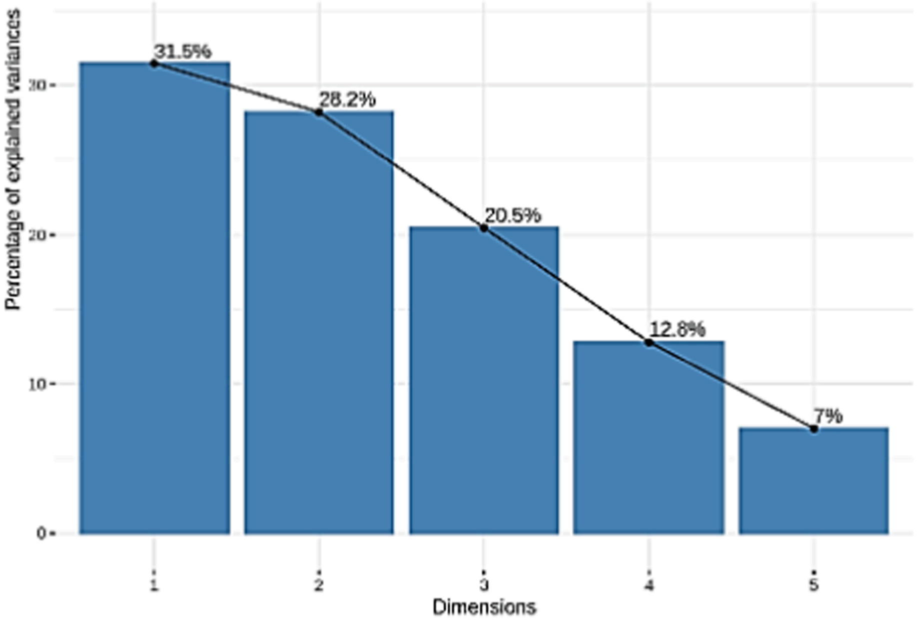

Fig. 4. Principle Component Analysis.

forms variable B more than variable B informs variable A. Technically, the score is a measurement in space [0,1] of the success of a model in predicting a variable target with the aid of an off-sample variable prediction, something which practically means that this method can increase the efficiency of finding hidden patterns in the data and selecting appropriate prediction variables. The final process of capturing the predictive power of the individual characteristics was done with the PPS technique, where for the calculation of PPS in numerical variables the metric of Mean Absolute Error (MAE) was used, which is the measure of quantification of error between estimation or prediction and the observed values. MAE is given by the Eq. (3):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i| = \frac{1}{n}\sum_{i=1}^{n}|e_i| \tag{3}$$

where f_i is the estimated value and y_i the true one. The average of the absolute value of the ratio between these values is defined as the absolute error of their relation $|e_i| = |f_i - y_i|$. The following Figs. 5, 6 and 7 illustrate the predictive power of the features used in the present investigation:

5. Methodology

In order to carry out a thorough investigation of the ML algorithms' ability to model the given problem based on the existing data, an extensive comparison of the most well-known algorithms was made. This comparison includes metrics for evaluating the performance of each algorithm, its execution time, generalization error, as well as the inherent behavior of the algorithm, in order to gain a deeper understanding of its theoretical basis.

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. Specifically, a comprehensive review of Support Vector Machines (SVMs) [65], Random Forest Classifier [66], CatBoost Classifier [67], Light Gradient Boosting Machine [68], Extreme Gradient Boosting [69], Extra Trees Classifier [70], Decision Tree Classifier [71], Gaussian Process Classifier [72], k-Neighbors Classifier [73], Linear Discriminant Analysis (LDA) [74], Ridge Classifier [75], Quadratic Discriminant Analysis (QDA) [76], MLP Classifier [77], Naive Bayes [78], AdaBoost Classifier [79] and Logistic Classifier [80].

5.1. Data sampling

In order to have an objective evaluation process of ML models, both as a way of self-evaluation and for their comparison with the corresponding alternative models, there are various statistical techniques of distribution and handling of datasets, which are also called validation techniques. K-Fold is the most common cross-validation method, according to which the dataset is randomly divided into k subsets, each of relatively equal population. Of the aforementioned k subsets, one is used 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 computing cycles is performed, so that, in turn, each of the k subsets is used as a test subset. The advantage of this evaluation method is that each data is used for training and definitely once for examination. The parameter k can attain any positive integer value, while the most popular choice in practical applications is the case where $k = 10$, which is called 10-Fold Cross Validation.

5.2. Comparison results

For the thorough evaluation of the buildings' seismic damage classification in the respective categories of damage level, a thorough investigation study was carried out with various machine ML. The flowchart of Fig. 8 shows how the ML algorithm that performs the best was selected. This flowchart clearly shows the different criteria used in the comparison.

We should mention that the following comparison is based on different performance metrics that are used to evaluate the different machine learning algorithms for the specific classification problem. Specifically, the classification performance metrics which used are Accuracy, Receiver Operating Characteristic (ROC), Recall, Precision, F-Score, Cohen's Kappa Statistics (CKS), and Matthews Correlation Coefficient (MCC). All metrics are the simple ratios between the number of correctly classified points to the total number of points, in the test dataset (unseen data points). The best performance is achieved by the model with a value near 1. The results by descent accuracy order for each dataset are presented in the following Tables 4, 5 and 6.

5.3. Best performance algorithm

In all three cases examined the Support Vector Machine (SVM) - Gaussian Kernel algorithm produced the highest classification results. The basic function of SVMs [65] is to construct a super-level that plays the role of a decision-making surface, so that the margin of separation of

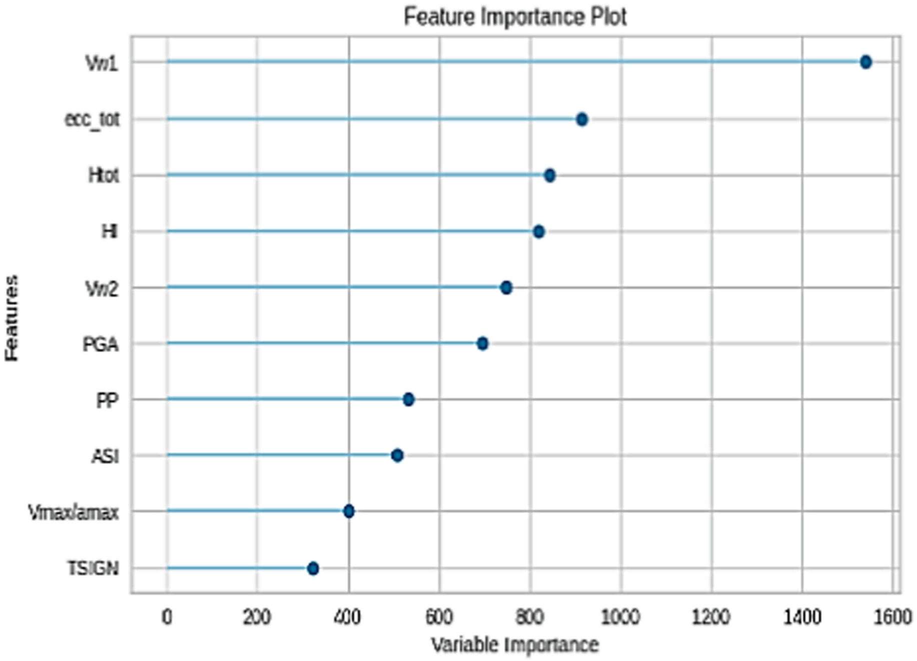

Fig. 5. Feature Importance plot for Row_Form_Bare dataset.

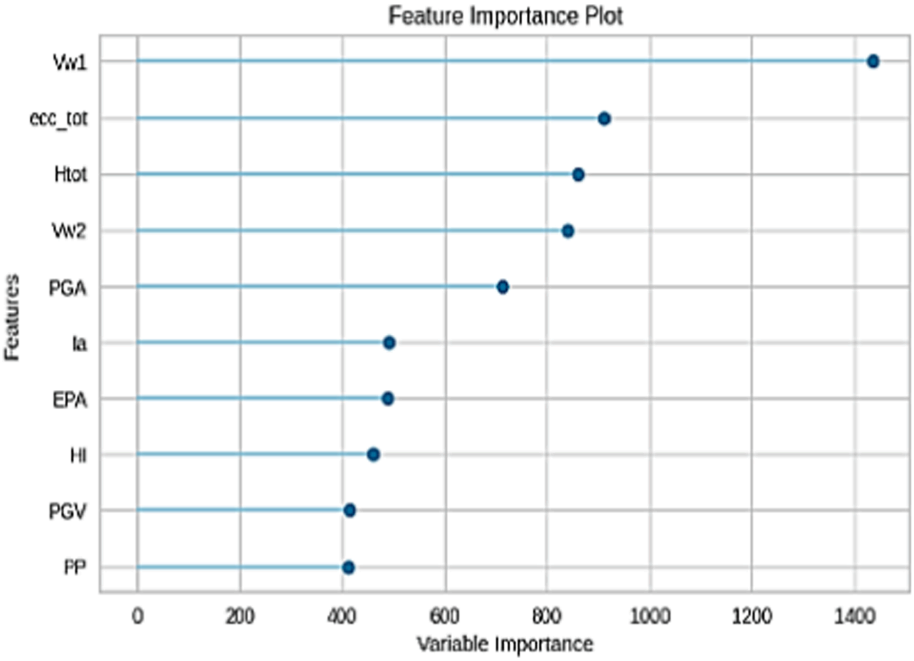

Fig. 6. Feature Importance plot for Row_Form_Full-Masonry dataset.

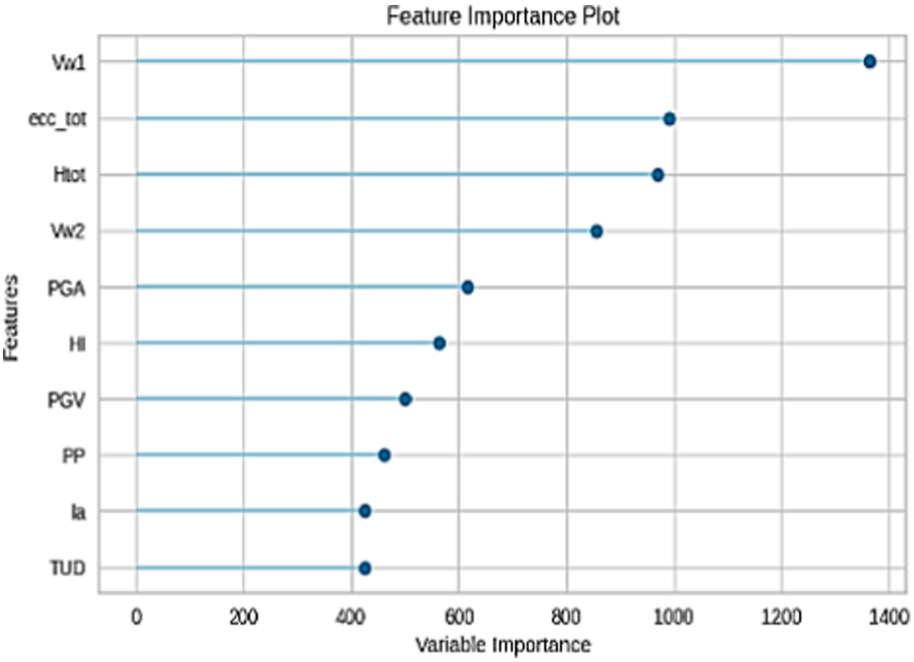

Fig. 7. Feature Importance plot for Row_Form_Pilotis dataset.

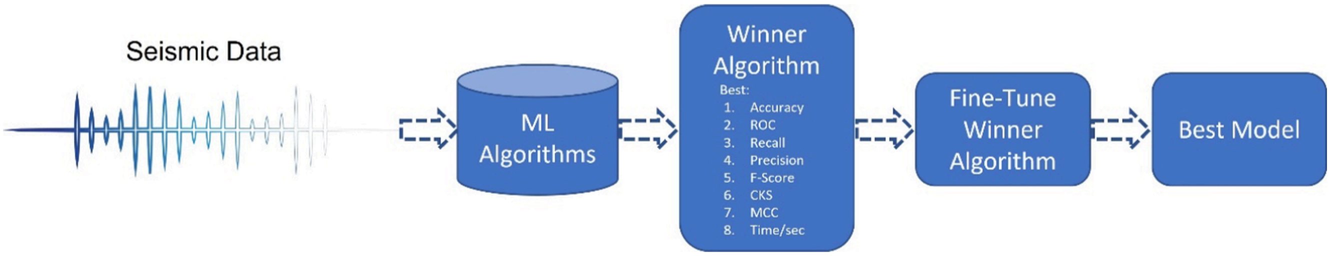

Fig. 8. Flowchart of the proposed process to find the best model.

Table 4
Performance Metrics in the Row_Form_Bare dataset.

ID	Model	Accuracy	ROC	Recall	Precision	F-Score	CKS	MCC	Time/sec
1	SVM - Gaussian Kernel	0.8849	0.9739	0.8683	0.8844	0.8843	0.8205	0.8208	0.659
2	Random Forest Classifier	0.8772	0.9691	0.8578	0.8756	0.8758	0.8081	0.8087	0.567
3	CatBoost Classifier	0.8757	0.9747	0.8553	0.8744	0.8744	0.8056	0.8063	4.158
4	Light Gradient Boosting Machine	0.8664	0.9724	0.8467	0.8659	0.8657	0.7915	0.7920	0.246
5	Extreme Gradient Boosting	0.8649	0.9716	0.8455	0.8654	0.8645	0.7892	0.7898	6.954
6	Extra Trees Classifier	0.8633	0.9635	0.8450	0.8630	0.8626	0.7868	0.7873	0.526
7	Decision Tree Classifier	0.8548	0.8915	0.8388	0.8573	0.8552	0.7744	0.7751	0.022
8	SVM - RBF Kernel	0.8471	0.9384	0.8228	0.8467	0.8449	0.7604	0.7623	0.359
9	Gaussian Process Classifier	0.8402	0.9130	0.8192	0.8407	0.8395	0.7508	0.7517	2.553
10	k-Neighbors Classifier	0.8224	0.9366	0.7992	0.8228	0.8213	0.7232	0.7245	0.120
11	Linear Discriminant Analysis	0.8124	0.9479	0.7920	0.8172	0.8134	0.7098	0.7111	0.021
12	SVM - Polynomial Kernel	0.8008	0.9308	0.7718	0.7968	0.7981	0.6885	0.6892	1.002
13	Ridge Classifier	0.7985	0.0000	0.7452	0.7859	0.7773	0.6785	0.6895	0.020
14	Quadratic Discriminant Analysis	0.7923	0.9419	0.7928	0.8186	0.7968	0.6854	0.6933	0.022
15	MLP Classifier	0.7483	0.9023	0.7383	0.7781	0.7486	0.6158	0.6277	0.320
16	Naive Bayes	0.7320	0.9205	0.7387	0.7820	0.7421	0.5992	0.6124	0.020
17	SVM - Gaussian Kernel	0.6797	0.8178	0.6552	0.7575	0.6826	0.5124	0.5369	0.146
18	Random Forest Classifier	0.6564	0.0000	0.5728	0.6130	0.5975	0.4246	0.4783	0.064

Table 5
Performance Metrics in the Row_Form_Full-Masonry dataset.

ID	Model	Accuracy	ROC	Recall	Precision	F-Score	CKS	MCC	Time/sec
1	SVM - Gaussian Kernel	0.8949	0.9777	0.8770	0.8970	0.8941	0.8197	0.8218	0.244
2	Random Forest Classifier	0.8942	0.9759	0.8745	0.8976	0.8935	0.8187	0.8211	15.896
3	CatBoost Classifier	0.8926	0.9763	0.8710	0.8950	0.8921	0.8154	0.8172	4.328
4	Light Gradient Boosting Machine	0.8918	0.9739	0.8685	0.8961	0.8918	0.8145	0.8169	0.562
5	Extreme Gradient Boosting	0.8864	0.9747	0.8635	0.8914	0.8861	0.8053	0.8082	0.665
6	Extra Trees Classifier	0.8834	0.9690	0.8577	0.8860	0.8830	0.7998	0.8018	0.516
7	Decision Tree Classifier	0.8749	0.8986	0.8550	0.8779	0.8743	0.7865	0.7887	0.021
8	SVM - RBF Kernel	0.8726	0.9498	0.8388	0.8765	0.8716	0.7806	0.7832	0.387
9	Gaussian Process Classifier	0.8687	0.9494	0.8336	0.8700	0.8666	0.7727	0.7753	0.128
10	k-Neighbors Classifier	0.8656	0.9290	0.8358	0.8705	0.8658	0.7707	0.7731	2.605
11	Linear Discriminant Analysis	0.8324	0.9442	0.7966	0.8390	0.8340	0.7134	0.7150	0.023
12	SVM - Polynomial Kernel	0.8184	0.9365	0.8051	0.8350	0.8235	0.6957	0.6994	0.021
13	Ridge Classifier	0.8015	0.0000	0.7338	0.7896	0.7840	0.6453	0.6568	0.019
14	Quadratic Discriminant Analysis	0.7837	0.8572	0.7465	0.8123	0.7898	0.6342	0.6411	0.149
15	MLP Classifier	0.7443	0.8879	0.6845	0.7293	0.7322	0.5531	0.5577	0.988
16	Naive Bayes	0.7405	0.9160	0.7636	0.7944	0.7525	0.5876	0.6040	0.020
17	SVM - Gaussian Kernel	0.6988	0.8728	0.6076	0.6877	0.6541	0.4719	0.5056	0.322
18	Random Forest Classifier	0.5476	0.0000	0.4867	0.5936	0.5150	0.2506	0.2980	0.067

Table 6
Performance Metrics in the Row_Form_Pilotis dataset.

ID	Model	Accuracy	ROC	Recall	Precision	F-Score	CKS	MCC	Time/sec
1	SVM - Gaussian Kernel	0.8795	0.9744	0.8518	0.8823	0.8792	0.8109	0.8125	6.119
2	Random Forest Classifier	0.8772	0.9754	0.8456	0.8776	0.8759	0.8067	0.8080	0.246
3	CatBoost Classifier	0.8772	0.9749	0.8433	0.8776	0.8752	0.8064	0.8082	4.387
4	Light Gradient Boosting Machine	0.8664	0.9710	0.8318	0.8673	0.8643	0.7895	0.7916	0.662
5	Extreme Gradient Boosting	0.8626	0.9677	0.8285	0.8658	0.8623	0.7841	0.7857	0.561
6	Extra Trees Classifier	0.8479	0.9622	0.8081	0.8505	0.8472	0.7607	0.7626	0.512
7	Decision Tree Classifier	0.8402	0.8820	0.8040	0.8432	0.8404	0.7493	0.7505	0.022
8	SVM - RBF Kernel	0.8370	0.9276	0.7900	0.8354	0.8326	0.7406	0.7440	0.363
9	Gaussian Process Classifier	0.8216	0.8974	0.7863	0.8269	0.8228	0.7207	0.7221	2.571
10	k-Neighbors Classifier	0.8162	0.9280	0.7772	0.8208	0.8161	0.7114	0.7135	0.122
11	Linear Discriminant Analysis	0.8162	0.8382	0.7897	0.8323	0.8213	0.7150	0.7179	0.146
12	SVM - Polynomial Kernel	0.8061	0.9375	0.7539	0.7991	0.8008	0.6927	0.6944	0.971
13	Ridge Classifier	0.8054	0.9495	0.7777	0.8212	0.8111	0.6987	0.7008	0.024
14	Quadratic Discriminant Analysis	0.7976	0.0000	0.7111	0.7769	0.7710	0.6705	0.6826	0.020
15	MLP Classifier	0.7807	0.9413	0.7845	0.8412	0.7962	0.6704	0.6851	0.020
16	Naive Bayes	0.7374	0.9003	0.6757	0.7679	0.7126	0.5884	0.6144	0.229
17	SVM - Gaussian Kernel	0.7305	0.9251	0.7327	0.8013	0.7470	0.5977	0.6156	0.020
18	Random Forest Classifier	0.6502	0.0000	0.5548	0.6278	0.5930	0.4302	0.4812	0.064

the categories is maximized. A key feature of SVMs that determines their function is the so-called support vectors, which consist a small subset of the training data used. Considering the problem of categorizing two categories, as described at one level in Fig. 9a, it is obvious that the two categories marked with the labels “+” and “o” are linearly separable. However, there are many lines ε_1, ε_2, ε_3, … which are multiple possible decision surfaces that can achieve the same result. The SVM algorithm seeks for the single line (ε^*) that separates the categories in such a way

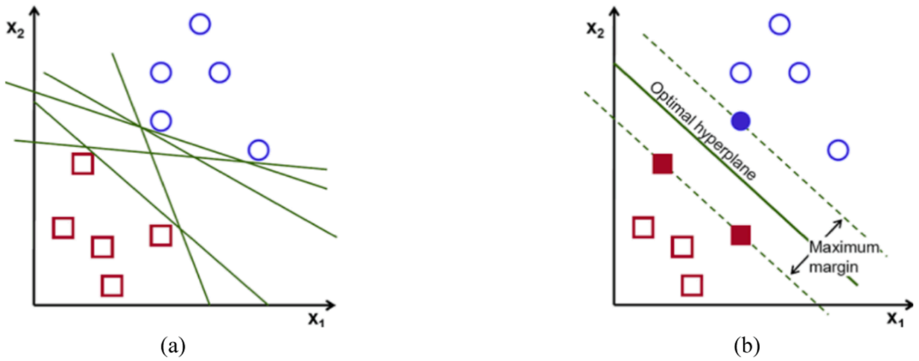

Fig. 9. Problem of two linearly separable categories.

that the margin between the categories is maximized, as shown in Fig. 9b, and consists the optimal decision surface.

In this case the data are linearly separable, something which guarantees the error-free classification of the data. As a consequence, the problem is reduced to the simplest case of patterns' classification, since the decision-making surface has the following simple form:

$$w^T x + b = 0 \tag{4}$$

where × is the input vector, *w* is the vector of the weights and *b* is the bias constant to be calculated. Because the data are linearly separable, the categorizer is described by the following Eqs. (5) and (6):

$$w^T x_k + b \geq +1, fort_k = +1 \tag{5}$$

or

$$w^T x_k + b \leq -1, fort_k = -1 \tag{6}$$

The above two equations can be described together, using the Eq. (7):

$$t_k\left(w^T x_k + b\right) \geq +1, k = 1, 2, 3, \cdots, N \tag{7}$$

The goal of SVMs is to find the decision-making surface by maximizing the margin that separates the categories, which equals to $\frac{2}{\|w\|_2}$. The vectors for which the equality of the above function applies are the so-called support vectors and are those vectors that lie closest to the decision-making surface and, therefore, those that are more difficult to categorize than all training vectors. Therefore, the problem of classification becomes an optimization problem, in which the optimal surface (w^*, b^*) that reduces the cost $J(w) = \frac{1}{2}w^T w$ satisfying some constraints is searched. This problem is defined as follows by the Eq. (8):

$$\frac{min}{w,b}\left\{J(w) = \frac{1}{2}w^T w\right\} \tag{8}$$

so that $t_k\left(w^T x_k + b\right) \geq +1, k = 1, 2, 3, \cdots, N$.

In the above optimization problem, which is called primal, the cost function is convex and the constraints are linear with respect to *w*. The solution is achieved with the aid of the Lagrange multipliers method, based on which the following Lagrange function is formed by Eq. (9):

$$L(w, b, a) = \frac{1}{2}w^T w - \sum_{k=1}^{N} a_k\left[t_k\left(w^T x_k + b\right) - 1\right] \tag{9}$$

where the coefficients $a_k \geq 0, k = 1, \cdots, N$ are called Lagrange multipliers.

The solution of the initial optimization problem with constraints becomes an *L(w,b,a)* saddle point optimization problem. In particular, this point should be maximized with respect to α and minimized with respect to *w* and *b*, by Eq. (10):

$$\frac{max}{a}\frac{min}{w,b}L(w, b, a) \tag{10}$$

Taking the derivatives of the function and setting them equal to zero, the following two Eqs. (11) and (12) arise:

$$\frac{\partial L(w, b, a)}{\partial w} = 0 \tag{11}$$

and

$$\frac{\partial L(w, b, a)}{\partial b} = 0 \tag{12}$$

From the two conditions of the function the following Eqs. (13) and (14) of a sigma point are derived:

$$w = \sum_{k=1}^{N} a_k t_k x_k \tag{13}$$

and

$$w = \sum_{k=1}^{N} a_k t_k = 0 \tag{14}$$

Substituting the above value of *w* into the function the dual optimization problem results, which is defined as follows by Eqs. (15) and (16):

$$\frac{min}{a}Q(a) = \sum_{k=1}^{N} a_k - \frac{1}{2}\sum_{l=1}^{N}\sum_{m=1}^{N} a_l a_m t_l t_m x_l^T x_m \tag{15}$$

and

$$\sum_{k=1}^{N} a_k t_k = 0\mu\varepsilon a_k \geq 0, k = 1, \cdots, N \tag{16}$$

The above becomes a Quadratic Programming problem, resulting in several non-zero a_k solutions which are the requested support vectors.

By finding the optimal Lagrange multipliers a_k^*, the weights w^* are calculated, while the corresponding bias b^* is determined from one of the data separation cases. In the opposite case (which is the most probable as most problems are non-linearly separable due to uncertainty, inaccuracy of representation and noise) there is a classification error, so the purpose of SVMs is to minimize this error. For this purpose, a new set of positive numbers called slack parameters is introduced, which measure the deviation of the data from the correct classification. In this case, the decision-making surface has the form of Eq. (17):

$$t_k\left(w^T x_k + b\right) \geq 1 - \xi_k, k = 1, 2, 3, \cdots, N \tag{17}$$

where $\xi_k \geq 0$ are the slack parameters, while the corresponding initial function optimization problem is transformed in the Eq. (18) as follows:

$$\frac{min}{w,b}\left\{J(w,\xi)=\frac{1}{2}w^{T}w+c\sum_{k=1}^{N}\xi_{k}\right\} \tag{18}$$

so that $t_k\left(w^T x_k + b\right) \geq 1 - \xi_k, \xi_k \geq 0, k = 1, 2, 3, \cdots, N$ where c is a positive constant which is usually determined experimentally. The corresponding Lagrange Eq. (19) will take the form:

$$L(w,b,\xi,a)=\frac{1}{2}w^{T}w+c\sum_{k=1}^{N}\xi_{k}-\sum_{k=1}^{N}a_{k}\left[t_{k}\left(w^{T}x_{k}+b\right)-1+\xi_{k}\right]-\sum_{k=1}^{N}v_{k}\xi_{k} \tag{19}$$

where $v_k \geq 0, k = 1, \cdots, N$ is a second (in addition to a_k) set of Lagrange multipliers.

In this case the sagmatic point optimization problem using slack parameters is described by Eq. (20) as follows:

$$\frac{max}{a,v}\ \frac{min}{w,b,\xi}L(w,b,\xi,a,v) \tag{20}$$

Finally, the problem of Quadratic Programming with slack parameters is defined by following Eqs. (21) and (22) as follows:

$$\frac{min}{a}Q(a)=\sum_{k=1}^{N}a_{k}-\frac{1}{2}\sum_{l=1}^{N}\sum_{m=1}^{N}a_{l}a_{m}t_{l}t_{m}x_{l}^{T}x_{m} \tag{21}$$

and

$$\sum_{k=1}^{N}a_{k}t_{k}=0,0\leq a_{k}\leq c,k=1,\cdots,N \tag{22}$$

with the additional restriction $a_k \leq c$. The bias b^* is calculated for those $a_k \leq c$ for which $\xi_k = 0$.

A major boost to the implementation of real problems was the development of nonlinear SVMs, which are based on the assumption that a nonlinearly separable pattern recognition problem can be transformed into a linearly separable one in a multidimensional space. The transformation from a space with few dimensions (input space) to a multidimensional space (feature space) can be achieved by applying a nonlinear mapping $\varphi(x)$. In this case, the decision-making surface is defined by Eq. (23) as follows:

$$\sum_{i=1}^{m}w_{i}\varphi_{i}(x)+b=0 \tag{23}$$

where m is the dimension of the whole set of the nonlinear transformations $\varphi(x)$, i.e. the dimension of the feature space, which is typically much larger than the dimension n of the input space. Assuming that $\varphi_0(x) = 1, \forall x, w_0 = b\kappa\alpha\iota\varphi(x) = [\varphi_0(x), \varphi_1(x), \cdots, \varphi_m(x)]^T$, the function can be written with the following form of the Eq. (24):

$$\sum_{i=1}^{m}w_{i}\varphi_{i}(x)=w^{T}\varphi(x)=0 \tag{24}$$

Considering that by using the mapping functions $\varphi(x)$ the problem has been reduced to a linear one with separable data in the space of the features, the solution of the Lagrange function for the whole set of the weights takes the form of the Eq. (25):

$$w=\sum_{k=1}^{N}a_{k}t_{k}\varphi(x_{k}) \tag{25}$$

And so, it is transformed into Eq. (26):

$$\sum_{k=1}^{N}a_{k}t_{k}\varphi^{\mathrm{T}}(x_{k})\varphi(x)=0 \tag{26}$$

The quantity $\varphi^{\mathrm{T}}(x_k)\varphi(x)$ describes the interior product of two vectors in the feature space. This quantity is called the kernel and is denoted by Eq. (27):

$$\mathrm{K}(x_{k},x)=\varphi^{\mathrm{T}}(x_{k})\varphi(x) \tag{27}$$

Based on Mercer's theorem, the kernel can be represented by Eq. (28) as:

$$\mathrm{K}(x_{k},x)=\sum_{i=0}^{m}\varphi_{i}(x_{k})\varphi_{i}(x),k=1,2,\cdots,N \tag{28}$$

a technique which is called kernel trick. A depiction of the kernel trick method is represented in Fig. 10.

Therefore, the decision-making surface will have the form of Eq. (29):

$$\sum_{k=1}^{N}a_{k}t_{k}K(x_{k},x)=0 \tag{29}$$

The corresponding dual quadratic programming optimization problem is defined by Eqs. (30) and (31) as follows:

$$\frac{min}{a}Q(a)=\sum_{k=1}^{N}a_{k}-\frac{1}{2}\sum_{l=1}^{N}\sum_{m=1}^{N}a_{l}a_{m}t_{l}t_{m}K(x_{l},x_{m}) \tag{30}$$

$$\sum_{k=1}^{N}a_{k}t_{k}=0,a_{k}\geq 0,k=1,\cdots,N \tag{31}$$

By finding the Lagrange multipliers from the above optimization problem the set of optimal weights w^* is calculated by the Eq. (32):

$$w^{*}=\sum_{k=1}^{N}a_{k}^{*}t_{k}\varphi(x_{k}) \tag{32}$$

where the first weight of the vector w^* corresponds to the optimal bias b^*.

It is worth noting that the selection of the appropriate kernel plays an important role in the performance of SVM. The only limitations that a kernel should satisfy are that the kernel must be symmetric.

The kernels that were used in the present investigation are the following (Eqs. (33), 34 and 35):

$$\mathrm{K}_{polynomial}(x_{k},x)=\left(\tau+x_{k}^{T}x\right)^{d} \tag{33}$$

$$\mathrm{K}_{RBF}(x_{k},x)=exp\left(-\frac{1}{2\sigma^{2}}\|x-x_{k}\|_{2}^{2}\right) \tag{34}$$

$$\mathrm{K}_{gaussian}(x_{k},x)=exp(-\gamma||u-v||2) \tag{35}$$

5.4. Results and discussion

When building and optimizing a classification model, measuring how accurately it predicts the expected outcome is crucial. However, one metric alone can offer misleading results. There are several performance evaluations metrics to help tease out more meaning in a model. The metrics to evaluate a machine learning model are very important as the choice of metrics influences how the performance of machine learning algorithms can be compared. Appropriate classification metrics were used in order to confirm the comparison results and to demonstrate the superiority of the machine learning algorithm that achieved the highest performance during the classification process. Specifically, given a pair of training vectors $\{x_i, y_i\}$, a classification model learns the parameters θ for an unknown function *f(x)*, which can match each input vector x_i to the estimated output *f(xi)*. Successful training means the optimal adaptation of the internal parameters θ, in order to minimize an error or cost function, which evaluates the performance of the categorizer based on some efficient and sound measures. The most popular evaluation measures, which are able to evaluate and compare with clarity, completeness and objectivity the classification algorithms are presented below [81,82]:

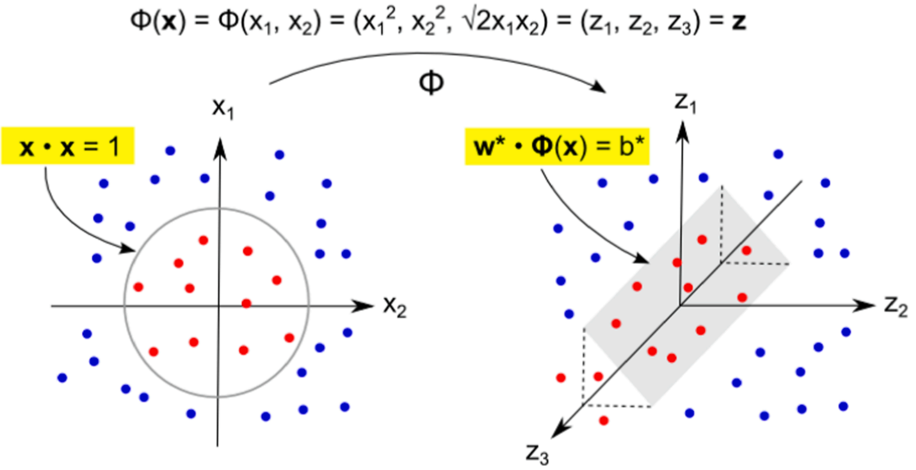

Fig. 10. Kernel Trick.

5.4.1 Confusion matrix

Because incorrect classifications of different classes have different costs, it is important to assess the predictor's ability to predict each class. To evaluate the performance for each class, the following terminology is used:

1. Positive: The observations which belong to a value of the class.
2. Negative: The observations which belong to the other value of the class.
3. True Positive (TP): The number of successful predictions for positive observations.
4. True Negative (TN): The number of successful predictions for negative observations.
5. False Positive (FP): The number of failed predictions for negative observations.
6. False Negative (FN): The number of failed predictions for positive observations.

The evaluation of a classification model is based on the number of records in the control set that are correctly or incorrectly predicted by the model. This number is placed in a confusion matrix (Table 7), which is a two-dimensional table, where the columns correspond to the predictions and the rows correspond to the actual values of the class. Table 8..

The principle of the Confusion Matrix is that it recognizes the nature of the errors, as well as their quantity. Each snapshot f_{ij} shows the number of records from class *i* that are expected to belong to class *j*. The f_{01} snapshot is the number of records from class 0 that were incorrectly predicted to be placed in class 1. Based on the snapshots, the number of records correctly predicted is the sum of f_{00} και f_{11}, while those predicted incorrectly are f_{01} and f_{10}. Although the Confusion Matrix provides the exact information needed to evaluate a model, this information can be expressed with aid of a unique number that is easy to use for comparisons between different models. Most performance measures can be expressed in relation to the number of TP, TN, FP and FN classifications for each class.

In the following Figs. 11, 12 and 13 are presented the confusion matrices of the seismic datasets by the SVM - Gaussian Kernel algorithm which produced the highest classification results. The Confusion Matrices visualizes the prediction score that takes a fitted classifier and a set of test X and y values and returns a report showing how each of the test values predicted classes compare to their actual classes. We use confusion matrices to understand which classes are most easily confused. Also, they provide deeper insight into the classification of individual data points.

Table 7
Confusion Matrix.

		Predicted Class	
		Class 1	Class 0
True Class	Class 1	f_{11} TP	f_{10} FN
	Class 0	f_{01} FP	f_{00} TN

The above confusion matrices show each combination of the true and predicted classes for each test data set. The true mode is selected, 100% accurate predictions are highlighted in green. Is the fact that the line from the True Positives which are at the top-left corner to True Negatives which are at the down-right corner, includes in the three confusion matrices most of the correct predictions in comparison to the actual values of the class.

5.4.2 Accuracy

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{36}$$

expresses the percentage of classification of control plots that are correctly categorized.

5.4.3 Precision

$$precision = \frac{TP}{TP + FP} \tag{37}$$

expresses the percentage of classification of the positive results that the categorizer has correctly classified as positive and are indeed positive. The higher the percentage of precision, the lower the corresponding percentage of FP.

5.4.4 Recall

$$recall = \frac{TP}{TP + FN} \tag{38}$$

expresses the percentage of classification of the positive examples that the categorizer was able to classify. The higher its percentage, the fewer positive examples have been incorrectly classified.

5.4.5 F-Score or F-measure or F1

In an attempt to objectively deal with cases where a categorizer has disproportionately distributed classification errors, the metric F-Score was introduced, which is the harmonic mean between precision and recall and is calculated:

Table 8
Performance Metrics by the SVM - Gaussian Kernel.

ID	Dataset	HS	Accuracy	ROC	Recall	Precision	F-Score	% Improvement
1	Row_Form_Bare	**BO**	**0.9113**	**0.9996**	**0.8906**	**0.9041**	**0.9017**	**2.98%**
2		GS	0.8947	0.9836	0.8770	0.8922	0.8914	1.07%
3		RS	0.9018	0.9905	0.8829	0.8974	0.8959	1.89%
4	Row_Form_Full-Masonry	**BO**	**0.9166**	**0.9990**	**0.8956**	**0.9137**	**0.9091**	**2.13%**
5		GS	0.9129	0.9956	0.8931	0.9117	0.9075	1.96%
6		RS	0.9132	0.9960	0.8933	0.9119	0.9076	2.01%
7	Row_Form_Pilotis	**BO**	**0.9055**	**0.9998**	**0.8734**	**0.9014**	**0.8960**	**2.96%**
8		GS	0.8941	0.9890	0.8645	0.8938	0.8895	1.61%
9		RS	0.8960	0.9908	0.8659	0.8949	0.8904	1.85%

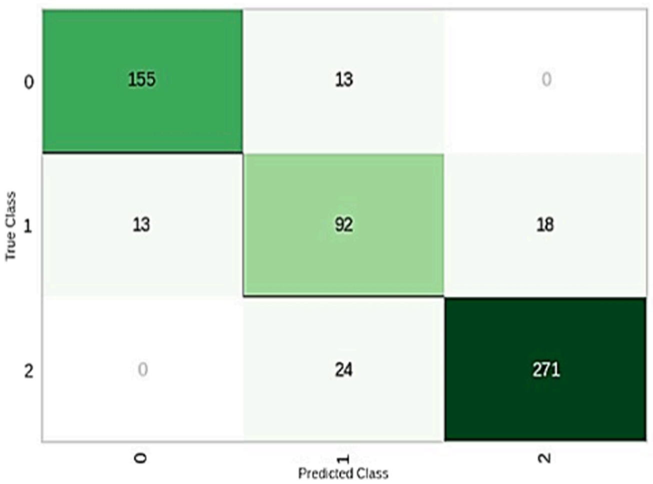

Fig. 11. Confusion Matrix of the Row_Form_Bare dataset (for classes' definition see Table 1).

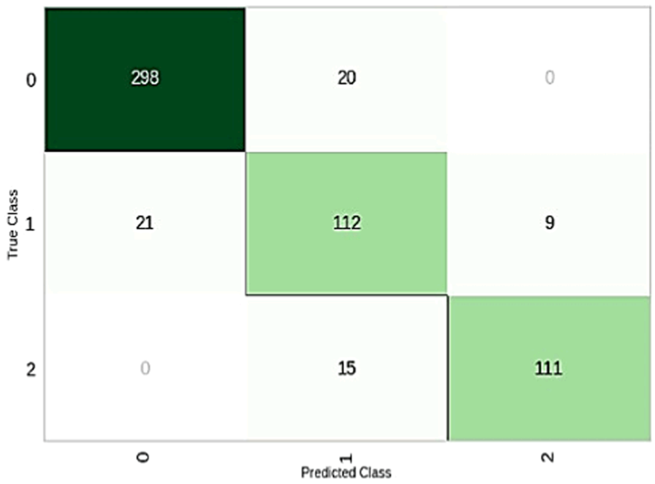

Fig. 12. Confusion Matrix of the Row_Form_Full-Masonry dataset (for classes' definition see Table 1).

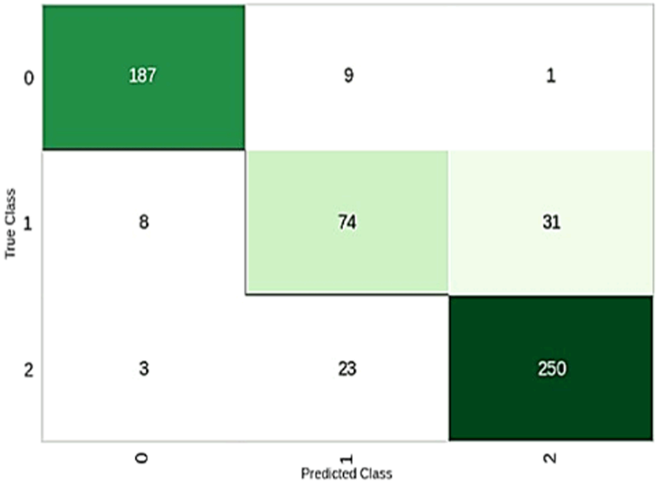

Fig. 13. Confusion Matrix of the Row_Form_Pilotis dataset (for classes' definition see Table 1).

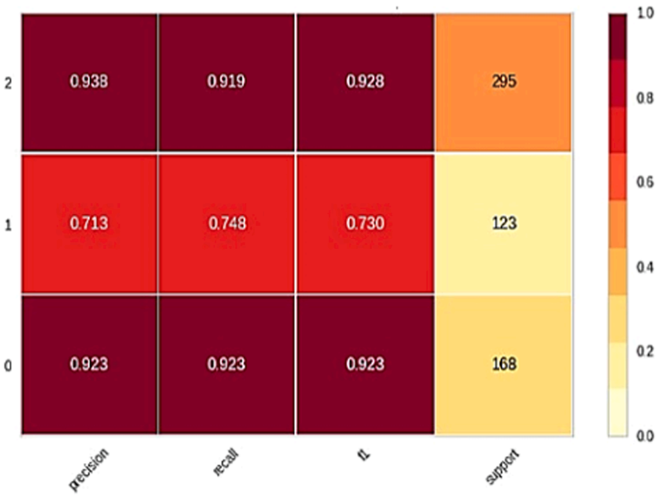

Fig. 14. Classification report of the Row_Form_Bare dataset.

$$F_{Score} = \frac{2 \times recall \times precision}{recall + precision} = \frac{2TP}{2TP + FP + FN} \tag{39}$$

The higher the percentage of the metric F-Score, the higher the respective two metrics. In the following Figs. 14, 15 and 16 are presented the classification reports plots by the SVM - Gaussian Kernel algorithm.

Each classification report visualizer displays the precision, recall, F1, and support scores for the model. In order to support easier interpretation and problem detection, the report integrates numerical scores with a color-coded heatmap. All heatmaps are in the range (0.0, 1.0) to facilitate easy comparison of classification models across different classification reports.

5.4.6 Receiver Operating Characteristic (ROC)

This metric can be applied to categorizers that have as output trust. In this case, the categorizer predicts one class if its confidence for it exceeds a threshold. For the formation of the ROC Curve, various threshold values are used and the True Positive Rate (TPR) and False Positive Rate (FPR) percentages are noted for each of them. These value

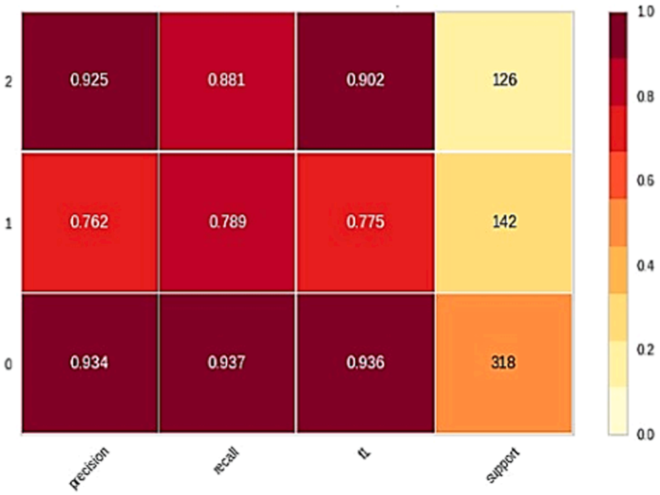

Fig. 15. Classification report of the Row_Form_Full-Masonry dataset.

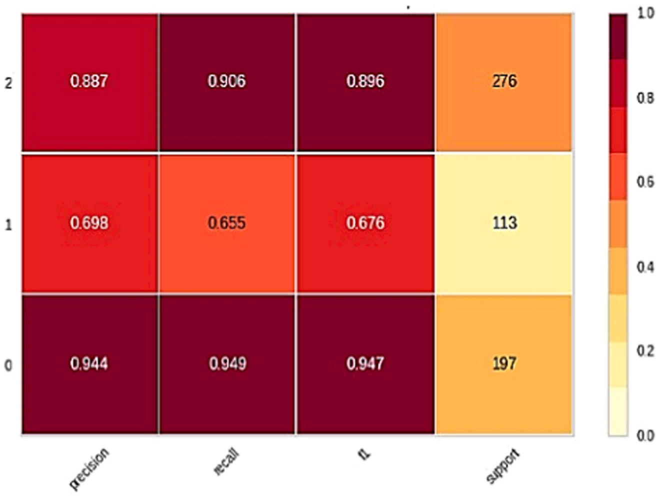

Fig. 16. Classification report of the Row_Form_Pilotis dataset.

pairs are plotted on a graph where the *y*-axis corresponds to the TPR and the *x*-axis to the FPR. The performance of each categorizer is represented by a point on the ROC curve. The advantages of this metric are that it gathers information about the prediction quality of the categorizer for different threshold values and is also independent of the class imbalance in the data. The following Figs. 17, 18 and 19 are presented the ROC curve plots by the SVM - Gaussian Kernel algorithm.

The above ROC curves are the measure of the SVM - Gaussian Kernel classifier predictive quality that compares and visualizes the tradeoff between the model’s sensitivity and specificity. The higher the ROC, the better the model generally is. However, it is also important to inspect the “steepness” of the curve, as this describes the maximization of the true positive rate while minimizing the false positive rate.

5.4.7. Error functions

Estimator function or estimator is a function of the random sample used to estimate an unknown parameter of a distribution function. The estimators in the case of classification refer to cost or error functions, which are able to quantify the classification variance achieved by an algorithm. The two cost or error functions used to categorize this method are the following:

1. Cohen’s Kappa Statistics (CKS): This is a statistical measurement that provides information about the amount of agreement between the

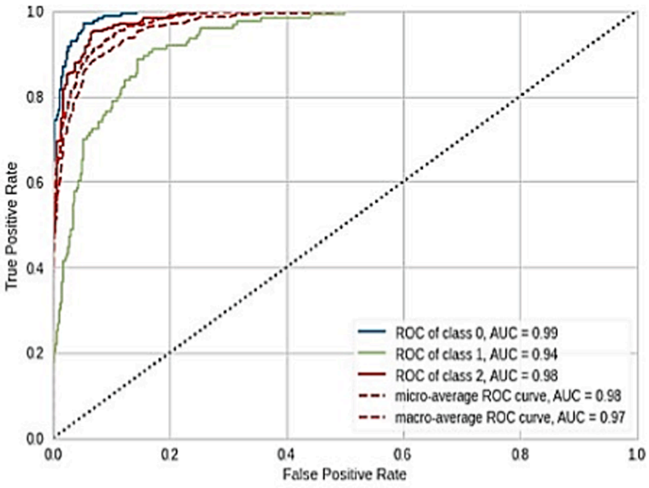

Fig. 17. ROC curve of the Row_Form_Bare dataset.

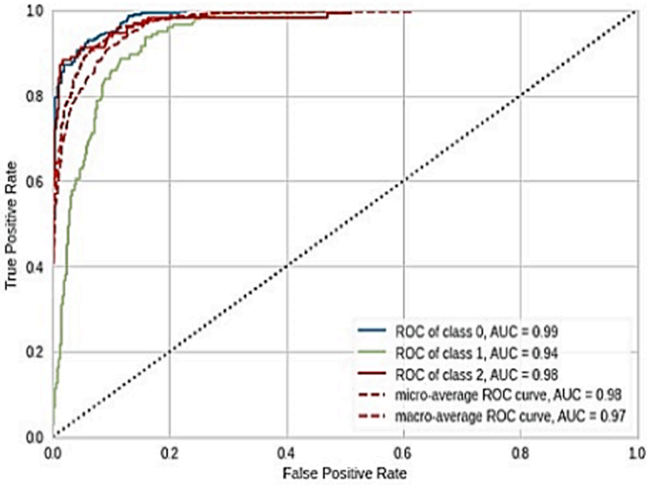

Fig. 18. ROC curve of the Row_Form_Full-Masonry dataset.

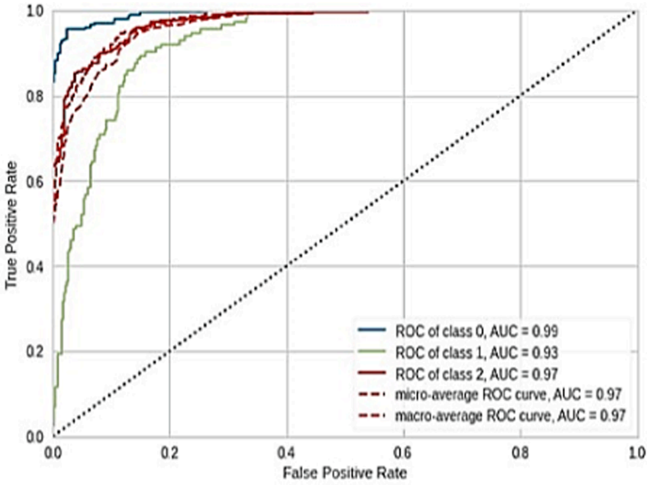

Fig. 19. ROC curve of the Row_Form_Pilotis dataset.

truth map and the final ranking map. It is the percentage agreement between two raters, where each classifies N items into C mutually exclusive categories. The definition of CKS calculate by the following Eq. 40:

$\kappa = \frac{p_0 - p_e}{1 - p_e} = 1 - \frac{1 - p_0}{1 - p_e}$ (40).

where p_o is the relative observed agreement among raters (identical to accuracy), and p_e is the hypothetical probability of chance agreement. The observed data are used to calculate the probabilities of each observer, to randomly see each category. More specifically, 0 = agreement equivalent to chance, 0.1 – 0.20 = slight agreement, 0.21 – 0.40 = fair agreement, 0.41 – 0.60 = moderate agreement, 0.61 – 0.80 = substantial agreement, 0.81 – 0.99 = near perfect agreement and 1 = perfect agreement.

2. Matthews Correlation Coefficient (MCC): MCC is used in machine learning as a measure of the quality of classifications. It is considered a balanced measure that can be used even if the sizes of the classes are very different, as it calculates a correlation coefficient value of [-1, +1] with + 1 representing a perfect prediction, 0 an average random prediction and − 1 a reverse prediction. It is calculated by the Eq. (41):

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}} \quad (41)$$

The class prediction error chart provides a way to quickly understand how good the classifier is at predicting the right classes. The following Class Prediction Error plots show the support (number of training samples) for each class in the fitted classification model as a stacked bar chart. Each bar is segmented to show the proportion of predictions (including false negatives and false positives, like the Confusion Matrix) for each class. We use each Class Prediction Error plot to visualize which classes of the classifier is having a particularly difficult time with, and more importantly, what incorrect answers it is giving on a per-class basis. This enables us to better understand the strengths and weaknesses of each model and particular challenges unique to each dataset. The Figs. 20, 21 and 22 are presented the Class Prediction Error plots by the SVM - Gaussian Kernel algorithm.

In the above example, while the classifier appears to be fairly good at correctly predicting classes 0 and 2 based on the features of the Row_Form_Bare dataset, it often incorrectly labels in class 1. Similar, in the above example, the classifier it often incorrectly labels in class 1. Also, in the above example, the classifier appears to be fairly good at correctly predicting class 2. Also, it often incorrectly labels in class 0 and 1.

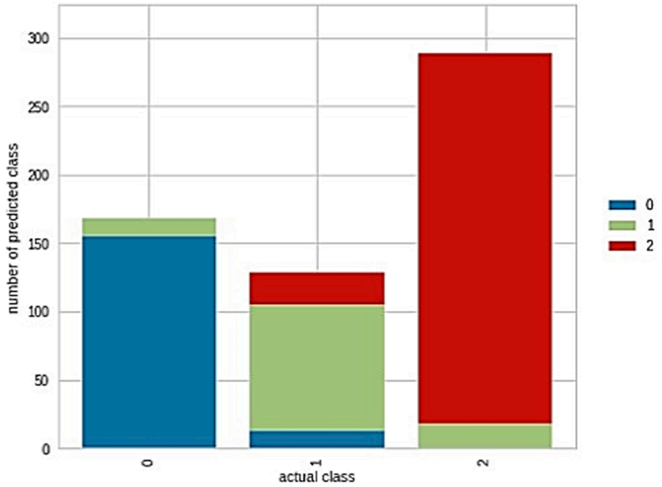

Fig. 20. Class Prediction Error of the Row_Form_Bare dataset (for classes' definition see Table 1).

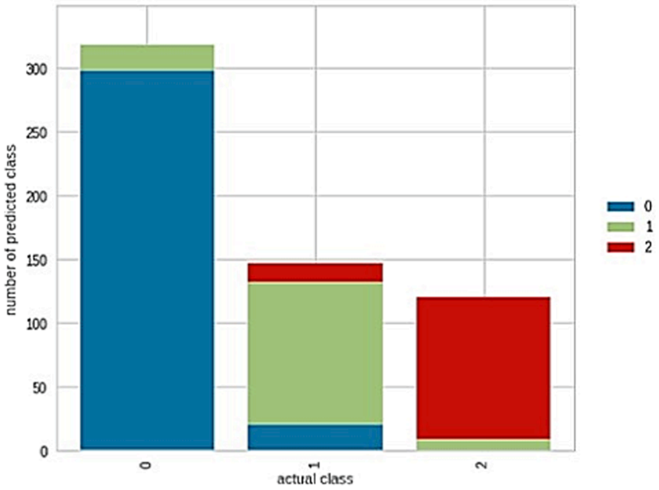

Fig. 21. Class Prediction Error of the Row_Form_Full-Masonry dataset (for classes' definition see Table 1).

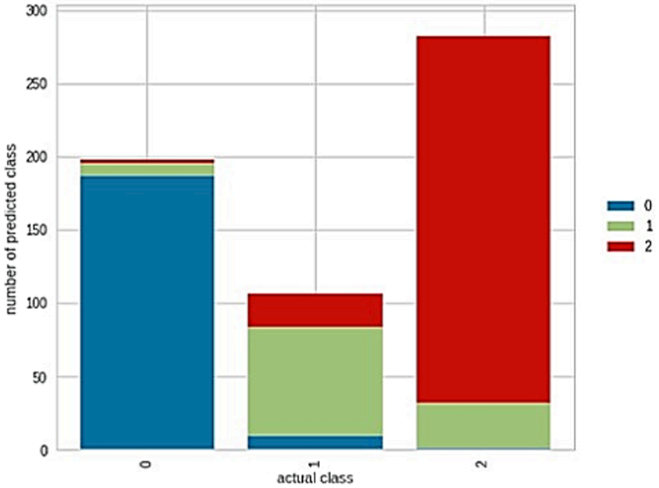

Fig. 22. Class Prediction Error of the Row_Form_Pilotis dataset (for classes' definition see Table 1).

6. Auto hyperparameter tuning

So, after the presentation of all the background material on Statistical Learning and how the Support Vector Machine model is generated, the automatic optimization of the hyperparameters utilizing Bayesian Optimization (BO) will be described in this section. Hyperparameters in machine learning are determined a priori rather than during training. In equation form, hyperparameter optimization is expressed as [83-86]:

$$x^{\star} = \underset{x \in X}{argmin} f(x) \quad (42)$$

In machine learning, hyperparameter optimization seeks to identify the hyperparameters of a given algorithm that produce the highest performance when compared against a validation set. The hyperparameters are the model settings that need to be tuned and, unlike model parameters, are set by the machine learning engineer before training. Here, f(x) is an objective score to minimize, such as RMSE or error rate, evaluated on the validation set and x* is the set of hyperparameters that provides the lowest score value (x can take any value in the domain X). We want to find the model hyperparameters that produce the highest score in test set. The issue with hyperparameter optimization is that evaluating the objective function to determine the score is expensive. Each time we experiment with alternative hyperparameters, we must train a model on training data, make predictions on validation data, and then calculate the validation measure. As a result, doing this process by hand gets increasingly difficult. Because we put up a grid of model hyperparameters and automatically conduct the train-predict-evaluate cycle in a loop, grid search and random search perform better than manual tweaking. However, even these methods are inefficient, since they do not select the next hyperparameters to examine based on previous results. Grid and random search are completely ignorant by previous evaluations and, as a result, frequently waste time considering “poor” hyperparameters.

When using the Gaussian kernel to train an SVM, two parameters must be considered: C and gamma. The parameter C, shared by all SVM kernels, trades off training example misclassification versus decision surface simplicity. A low C smoothest the decision surface, whereas a high C attempts to identify all training samples correctly. Gamma is the amount of influence that a single training example has. The greater the gamma, the closer the other instances must be to be influenced. The selection of C and gamma is essential to the performance of the SVM. To find suitable values, optimizing with C is recommended, and gamma spreads exponentially apart. The most often used method for tweaking hyperparameters in learning models is grid search. This heuristic technique tests many parameter possibilities before selecting the one that best matches the introduced fitted data. However, it is inefficient due to the many combinations to try. It is impossible to decide whether the variety discovered is the best even after a thorough search. This paper uses training with BO approaches to overcome this impact.

The fundamental idea behind Bayesian model-based optimization is to limit the number of times the objective function must be run by evaluating only the most promising set of hyperparameters based on past calls to the evaluation function. The next set of hyperparameters is chosen based on a surrogate model of the objective function. The surrogate function is a probability representation of the objective function constructed from past evaluations. Because it is a high-dimensional mapping of hyperparameters to the probability of a score on the objective function, this is sometimes referred to as a response surface. The selection function is the criteria used to select the next set of hyperparameters from the surrogate function. Expected Improvement (EI) is the most commonly used criterion:

$$\mathrm{EI}_{y^*}(x) = \int_{-\infty}^{y^*} (y^* - y) p(y|x) dy \tag{43}$$

Here, y* is the objective function’s threshold value, x is the suggested set of hyperparameters, y is the objective function’s actual value employing hyperparameters ×, and p(y|x) is the surrogate probability model representing the probability of y given x. The goal is to maximize the Expected Improvement to x. Finding the optimum hyperparameters under the function p(y|x) is what this entails. If p(y|x) is zero everywhere that y < y*, then the hyperparameters × are unlikely to improve. If the integral is positive, the hyperparameters × are projected to outperform the threshold value. The BO, instead of directly representing p(y|x), employs the following:

$$(y|x) = \frac{p(x|y)^* p(y)}{p(x)} \tag{44}$$

where p(x|y) is the probability of the hyperparameters given the objective function score, represented as:

$$p(x|y) = \begin{cases} l(x) \text{ if } y < y^* \\ g(x) \text{ if } y \geq y^* \end{cases} \tag{45}$$

where y < y* represents a lower objective function value than the threshold. This equation explains that we make two different hyperparameter distributions: the objective function value is less than the threshold l(x), and the objective function value is greater than the threshold g(x). From this perspective, the predicted EI (which we are attempting to optimize) is:

$$EI_{y^*}(x) = \frac{\gamma y^* l(x) - l(x) \int_{-\infty}^{y^*} p(y) dy}{\gamma l(x) + (1-\gamma) g(x)} \propto \left(\gamma + \frac{g(x)}{l(x)}(1-\gamma)\right)^{-1} \tag{46}$$

This means that the EI is proportional to the ratio l(x)/g(x), and hence we should increase this ratio to maximize the EI. As a result, we should choose hyperparameter values that are more likely under l(x) than under g(x). Because l(x) is a distribution rather than a single value, the hyperparameters chosen are likely to be close but not strictly at the maximum of the projected improvement. Furthermore, because the surrogate is only an approximation of the objective function, the chosen hyperparameters may not result in a gain when evaluated, and the surrogate model will need to be modified. The present surrogate model and the history of objective function evaluations are used to update this model. The proposed methodology uses Bayesian Inference (BI) to address this issue. BI is a statistical inference approach that uses Bayes’ theorem to update the probability of a hypothesis as new data or information becomes available. The posterior probability is calculated by BI using two antecedents: a prior probability and a “likelihood function” obtained from a model for the observed data. BI calculates the posterior probability using Bayes’ theorem:

$$P(H|E) = \frac{P(E|H) \cdot P(H)}{P(E)} \tag{47}$$

The posterior probability is proportional to its prior probability and the newly acquired likelihood. Only the components P(H) and P(E | H) (both in the numerator) influence the value of P(H | E) for different values of H. Bayes’ rule can alternatively be written as:

$$\begin{aligned} P(H|E)\& &= \frac{P(E|H)P(H)}{P(E)} = \frac{P(E|H)P(H)}{P(E|H)P(H) + P(E|\neg H)P(\neg H)} \\ &= \frac{1}{1 + \left(\frac{1}{P(H)} - 1\right) \frac{P(E|\neg H)}{P(E|H)}} \end{aligned} \tag{48}$$

Because

$$P(E) = P(E|H)P(H) + P(E|\neg H)P(\neg H) \tag{49}$$

And

$$P(H) + P(\neg H) = 1 \tag{50}$$

Where ¬H is not H. Based on Rule of Multiplication:

$$P(E \cap H) = P(E|H)P(H) = P(H|E)P(E) \tag{51}$$

The prior distribution is the distribution of the parameters prior to the observation of any data, $p(\theta|\alpha)$.

The sampling distribution is the observed data distribution conditional on its parameters, $p(\mathrm{X}|\theta)$.

The marginal likelihood is the minimized distribution of observed data over the parameters:

$$p\left(\mathrm{X}\middle|\alpha\right) = \int p\left(\mathrm{X}\middle|\theta\right) p\left(\theta\middle|\alpha\right) d\theta \tag{52}$$

The posterior distribution is the distribution of the parameters after taking the observed data into account:

$$p(\theta|\mathrm{X},\alpha) = \frac{p(\theta,\mathrm{X},\alpha)}{p(\mathrm{X},\alpha)} = \frac{p(\mathrm{X}|\theta,\alpha)p(\theta,\alpha)}{p(\mathrm{X}|\alpha)p(\alpha)} = \frac{p(\mathrm{X}|\theta,\alpha)p(\theta|\alpha)}{p(\mathrm{X}|\alpha)} \propto p(\mathrm{X}|\theta,\alpha)p(\theta|\alpha) \tag{53}$$

So, the distribution of a new data point, marginalized over the posterior, is the posterior predictive distribution:

$$p(\widetilde{x}|\mathrm{X},\alpha) = \int p(\widetilde{x}|\theta) p(\theta|\mathrm{X},\alpha) d\theta \tag{54}$$

And finally, the prior predictive distribution is a new data point that has been marginalized over the prior distribution:

$$p(\widetilde{x}|\alpha) = \int p(\widetilde{x}|\theta)p(\theta|\alpha)d\theta \quad (55)$$

In order to prove the ability of the proposed Bayesian method against random search, we have conducted a sophisticated experiment. Using the datasets which are into consideration in this paper, we have tuned the SVM Gaussian Kernel model using Hyperparameter Search (HS). Specifically, we used the proposed BO, Grid Search (GS) and Random Search (RS) models in 20 iterations for all and the target is to minimizing RMSE. The process was repeated three times with different seeds and the results averaged. Below is the table with comparison results.

As proved by the experiments, the proposed BO method maintains track of all assessments and utilizes the data to build a "surrogate probability model" that the machine learning model, the SVM Gaussian Kernel in this approach, can evaluate faster. Specifically, instead of simply sampling random configurations, the algorithm can choose which hyperparameters to assess next based on the prior and posterior distributions of the Bayesian Inference. In essence, this method creates a probability model of the objective function and then utilizes it to select the most promising hyperparameters to evaluate the objective function. The more combinations considered, the more informed the algorithm becomes, and the surrogate model becomes closer to the objective function.

7. Conclusions

The present paper investigates the classification of buildings' potential for seismic damage using a machine learning model with auto hyperparameter tuning. For this aim, a training dataset consisting of 30 3D R/C buildings with different structural parameters was chosen. The buildings were designed based on the provisions of EC8 and EC2. For each one of these buildings three different configurations as regards their masonry infills were considered (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 with 30 buildings each. Then, the buildings were analyzed my means of the Nonlinear Time History Analyses method for 65 appropriately chosen real earthquake records. Both seismic and structural parameters widely used in the literature were selected as inputs in the process of Machine Learning methods. The quantification of the buildings' damage level was done by means of the well-documented Maximum Interstory Drift Ratio. As proved, the Machine Learning methods that are mathematically well-established and their operations that are clearly interpretable step by step can be used to solve some of the most sophisticated real-world problems in consideration. The findings of the present work can be summarized as follows:

1. Different Machine Learning algorithms running on the same dataset can lead to different predictions with little or no overlap between them and, also, different parameters of the same algorithm can affect the results of the buildings' damage assessment.
2. The SVM method used is not prone to overfitting to a specific data set compared to other methods and is also a robust method against data noise and outliers.
3. The important advantage of the SVM is that the optimization method that is used by the algorithm presents a total minimum, giving a unique optimal choice, which does not happen in other methods such as Neural Networks that can be trapped in local minima.
4. The performance of the SVM classifier depends not only on the algorithmic mechanism of the classifier itself (decision method) but also on the kernel to be applied to it, as its performance varies with the use of different kernels. Adjusting a kernel suitable to improve alignment with the samples of the training set which are labeled, significantly increases the fit with the samples of the test set, giving quite improved classification accuracy. Therefore, choosing the right kernel is a vital issue for the performance of the final classification model.
5. The SVMs have significant generalizability to non-linearly separable data by incorporating the kernel trick. By applying kernel functions, it is possible to produce nonlinear models that lead to linearity in larger spaces. In addition, the number of parameters to be configured in SVMs is smaller compared to several corresponding methodologies. Also, for the classification of a new element in a class, the classification process is based only on the similarity of the element unknown to the algorithm and the most important elements of each class (support vectors), so the method reduces significantly the computational cost and the requirement resources.
6. The proposed approach generates an auto hyperparameter tuning model of the SVM algorithm, which is then used to simplify the ML process by finding the most promising hyperparameters for evaluating the used dataset.
7. Instead of picking random configurations, the algorithm decides which hyperparameters to estimate next depending on the Bayesian Inference's prior and posterior distributions.
8. The results confirm the need for further and in-depth exploration of the results related to the thorough evaluation of similarity measures and/or distance functions used in statistical approaches to implement features of the similar issues, in order to reduce the uncertainty in the decisions that include the data used to predict damages.
9. Because the approach adopted is mathematically well-interpretable, it can persuade stakeholders to trust and deploy intelligent technologies in the civil engineering sector.

Future work on this topic could focus on applying all of the above learning and prediction methods/models to different data sets, in order to assess whether they perform equally well on different data. This is also a way to test the insides of other methods of experts in terms of their robustness, i.e. to compare their performance for different data and to see if their accuracy is around the same percentages, so that they can be considered reliable as models. Also, ensemble methods could be applied with different combinations of individual classifiers, in terms of the number and nature of the latter, for further analysis of their function. In addition, other Machine Learning or Deep Learning models could be considered, such as the method based on the expectation–maximization algorithm, LSTM Neural Networks, etc. Finally, it will be important to look at different ways of selecting features, as well as mixing different selection methods, where it is expected very little overlap of insides indicators, in order to see if mixing methodology can statistically or mechanically lead to increased performance of the methods.

CRediT authorship contribution statement

Konstantinos Kostinakis: Software. **Konstantinos Morfidis:** Software. **Konstantinos Demertzis:** Software. **Lazaros Iliadis:** Supervision.

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

Data will be made available on request.

References

[1] FEMA P-154. Rapid visual screening of buildings for potential seismic hazards: a handbook (3rd Edition), Homeland Security Department, Federal Emergency Management Agency: Washington, DC, USA; 2015.

[2] Fema-356.. Prestandard and commentary for the seismic rehabilitation of buildings. American Society of Civil Engineers; 2000.

[3] Nrc-cnrc.. Manual for screening of buildings for seismic investigation, Institute for Research in Construction. Ottawa: National Research Council Canada; 1993.

[4] First Stage Rapid Visual Screening - Greek Rapid Visual Investigation Methodology (TOE) 5th edition, Earthquake Planning and Protection Organization of Greece (E. P.P.O.), 2020. Available at: https://www.oasp.gr/node/74.

[5] En1998-3.. Design of structures for earthquake resistance - part 3: Assessment and retrofitting of buildings. European Committee for Standardization 2005.

[6] Japan Building Disaster Prevention Association (JPDPA), Seismic evaluation and retrofit, Japan; 2001.

[7] New Zealand Society for Earthquake Engineering (NZSEE). Assessment and Improvement of the Structural Performance of Buildings in Earthquakes; Recommendations of a NZSEE Study Group on Earthquake Risk Buildings, June 2006; NZSEE: Wellington, New Zealand, 2006.

[8] Gndt. Seismic Risk of public buildings - Part 1 - Methodology Aspects (in Italian). Rome, Italy: CNR; 1993.

[9] Kappos AJ, Panagopoulos G, Panagiotopoulos Gr. Ch. Penelis, A hybrid method for the vulnerability assessment of R/C and URM buildings. Bull Earthquake Eng 2006; 4:391–413. https://doi.org/10.1007/s10518-006-9023-0.

[10] Anagnos T, Rojahn Ch, Kiremidjian A. NCEER-ATC joint study on fragility of buildings, Technical Report NCEER 95–0003. National Center for Earthquake Engineering Research: State Univ. of New York at Buffalo; 1995.

[11] Bayat M, Kosarieh AH, Javanmard M. Probabilistic seismic demand analysis of soil nail wall structures using bayesian linear regression approach. Sustainability 2021; 13(11):5782.

[12] Kia M, Banazadeh M, Bayat M. Rapid seismic loss assessment using new probabilistic demand and consequence models. B Earthq Eng 2019;17(6):3545–72.

[13] Bayat M, Emadi A, Kosariyeh AH, Kia M, Bayat M. Collapse fragility analysis of the soil nail walls with shotcrete concrete layers. Comput Concr 2022;29(5):279–83.

[14] Ningthoujam MC, Nanda RP. Rapid visual screening procedure of existing building based on statistical analysis. Int J Disaster Risk Reduct 2018;28:720–30. https://doi.org/10.1016/j.ijdrr.2018.01.033.

[15] Cremen G, Baker JW. Improving FEMA P-58 non-structural component fragility functions and loss predictions. Bull Earthquake Eng 2019;17(4):1941–60.

[16] Ramamoorthy KS, Gardoni P, Bracci MJ. Probabilistic demand models and fragility curves for reinforced concrete frames. ASCE J Struct Eng 2006;132(10):1563–72.

[17] Sfahani MG, Guan H, Loo Y-C. Seismic reliability and risk assessment of structures based on fragility analysis - A review. Adv Struct Eng 2015;18(10):1653–69. https://doi.org/10.1260/1369-4332.18.10.1653.

[18] Miano A, Mele A, Prota A. Fragility curves for different classes of existing RC buildings under ground differential settlements. Eng Struct 2022;257:114077. https://doi.org/10.1016/j.engstruct.2022.114077.

[19] Coskun O, Aldemir A, Sahmaran M. Rapid screening method for the determination of seismic vulnerability assessment of rc building stocks. Bull Earthq Eng 2020;18 (4):1401–16.

[20] Harirchian E, Hosseini SEA, Jadhav K, Kumari V, Rasulzade S, Işık E, et al. A review on application of soft computing techniques for the rapid visual safety evaluation and damage classification of existing buildings. J Build Eng 2021;43: 102536.

[21] Xie Y, Ebad SM, Padgett JE, DesRoches R. The promise of implementing machine learning in earthquake engineering: A state-of-the-art review. Earthq Spectra 2020; 36(4):1769–801.

[22] Sun H, Burton HV, Huang H. Machine learning applications for building structural design and performance assessment: State - of - the - art review. J Build Eng 2021; 33:101816.

[23] Rafiq MY, Bugmann G, Easterbrook DJ. Neural network design for engineering applications. Comput Struct 2001;79:1541–52.

[24] Aoki T, Ceravolo R, De Stefano A, Genovese C, Sabia D. Seismic vulnerability assessment of chemical plants through probabilistic neural networks. Reliab Eng Syst Saf 2002;77(3):263–8.

[25] Lautour OR, Omenzetter P. Prediction of seismic-induced structural damage using artificial neural networks. Eng Struct 2009;31:600–6.

[26] Tesfamariam S, Liu Z. Earthquake induced damage classification for reinforced concrete buildings. Struct Saf 32(2), pp. 154-164.

[27] Arslan MH, Ceylan M, Koyuncu T. Determining earthquake performances of existing reinforced concrete buildings by using ANN. Int J Civil, Environ, Struct, Constr Archit Eng 2015;9(8):930–4.

[28] Kia A, Sensoy S. Classification of earthquake-induced damage for R/C slab column frames using multiclass SVM and its combination with MLP neural network. Math Problems in Eng 2014b; 2014: 734072.

[29] Morfidis K, Kostinakis K. Seismic parameters' combinations for the optimum prediction of the damage state of R/C buildings using neural networks. Adv Eng Softw 2017;106:1–16.

[30] Morfidis K, Kostinakis K. Use of artificial neural networks in the R/C buildings' seismic vulnerability assessment: the practical point of view. In: Proceedings of COMPDYN2019, Crete island, Greece, 24-26; June 2019.

[31] Morfidis K, Kostinakis K. Approaches to the rapid seismic damage prediction of r/c buildings using artificial neural networks. Eng Struct 2018;165:120–41.

[32] Morfidis K, Kostinakis K. Rapid prediction of seismic incident angle's influence on the damage level of RC buildings using artificial neural networks. Appl Sci 2022. https://doi.org/10.3390/app12031055.

[33] Morfidis KE, Kostinakis KG. Approach to prediction of R/C buildings' seismic damage as pattern recognition problem using artificial neural networks, In: Proceedings of 4th Conference in Computational Methods in Structural Dynamics and Earthquake Engineering (COMPDYN2017), Rhodes island, Greece; 2017.

[34] Morfidis K, Kostinakis K. Comparative evaluation of MFP and RBF neural networks' ability for instant estimation of r/c buildings' seismic damage level. Eng Struct 2019;197.

[35] Zhang Y, Burton HV, Sun H, Shokrabadi M. A machine learning framework for assessing post-earthquake structural safety. Struct Saf 2018;72:1–16.

[36] Zhang Y, Burton HV. Pattern recognition approach to assess the residual structural capacity of damaged tall buildings. Struct Saf 2019;78:12–22.

[37] Harirchian E, Lahmer T. Developing a hierarchical type-2 fuzzy logic model to improve rapid evaluation of earthquake hazard safety of existing buildings. Structures 2020;28:1384–99.

[38] Mangalathu S, Sun H, Nweke CC, Yi Z, Burton HV. Classifying earthquake damage to buildings using machine learning. Earthq Spectra 2020;36(1):183–208.

[39] Guan X, Burton H, Sabol T. Python-based computational platform to automate seismic design, nonlinear structural model construction and analysis of steel moment resisting frames. Eng Structures 2020;224:111199.

[40] Huang H, Burton HV. Classification of in-plane failure modes for reinforced concrete frames with infills using machine learning. J Build Eng 2019;25:100767.

[41] Sun H, Burton H, Wallace J. Reconstructing seismic response demands across multiple tall buildings using kernel-based machine learning methods. Struct Control Health Monit 2019;26(7):e2359.

[42] Harirchian E, Lahmer T. Improved rapid assessment of earthquake hazard safety of structures via artificial neural networks. In: IOP conference series: materials science and engineering, vol. 897. IOP Publ; 2020. p. 012014.

[43] Harirchian E, Lahmer T, Rasulzade S. Earthquake hazard safety assessment of existing buildings using optimized multi-layer perceptron neural network. Energies 2020;13(8):2060.

[44] Harirchian E, Kumari V, Jadhav K, Rasulzade S, Lahmer T. A machine learning framework for assessing seismic hazard safety of reinforced concrete buildings. Applied Sciences (Swi) 2020;10(20).

[45] Harirchian E, Lahmer T, Kumari V, Jadhav K. Application of support vector machine modeling for the rapid seismic hazard safety evaluation of existing buildings. Energies 2020;13(13):3340.

[46] Harirchian E, Kumari V, Jadhav K, Rasulzade S, Lahmer T, Das RR. A synthesized study based on machine learning approaches for rapid classifying earthquake damage grades to rc buildings. Appl Sci (Switzerland) 2021;11(16):7540.

[47] Avci O, Abdeljaber O, Kiranyaz S. Structural damage detection in civil engineering with machine learning: current state of the art. In: Sensors and Instrumentation, Aircraft/Aerospace, Energy Harvesting & Dynamic Environments Testing; Springer: Cham, Switzerland, pp. 223-229, 2022.

[48] Wang C, Song LH, Fan JS. End-to-End Structural analysis in civil engineering based on deep learning. Autom Constr 2022;138:104255.

[49] Van-Thien Tran, Trung-Kien Nguyen, H. Nguyen-Xuan, Magd Abdel Wahab, Vibration and buckling optimization of functionally graded porous microplates using BCMO-ANN algorithm. Thin-Wall Struct 2023;Vol. 182, Part B, 110267. https://doi.org/10.1016/j.tws.2022.110267.

[50] Duong Huong Nguyen. Magd Abdel Wahab, Damage detection in slab structures based on two-dimensional curvature mode shape method and Faster R-CNN. Adv Eng Softw 2023;176:103371. https://doi.org/10.1016/j.advengsoft.2022.103371.

[51] Mohammad Rezaul Karim, Kamrul Islam, A.H.M. Muntasir Billah, and M. Shahria Alam, Shear Strength Prediction of Slender Concrete Beams Reinforced with FRP Rebar Using Data-Driven Machine Learning Algorithms, Journal of Composites for Construction, 27(2), 2023.

[52] EN1992-1-1. Design of concrete structures, Part 1-1: General rules and rules for buildings. European Committee for Standardization; 2005.

[53] EN1998-1. Design of structures for earthquake resistance - part 1: general rules, seismic actions and rules for buildings, European Committee for Standardization; 2005.

[54] Crisafulli FJ. Seismic behaviour of reinforced concrete structures with masonry infills. Christchurch, New Zealand: University of Canterbury; 1997. Ph.D. Thesis.

[55] PEER (Pacific Earthquake Engineering Research Centre). Strong motion database, 2003: https://ngawest2.berkeley.edu/.

[56] European Strong-Motion Database, 2003: http://isesd.hi.is/ESD_Local/frameset.htm.

[57] Carr AJ. Ruaumoko - A program for inelastic time-history analysis: program manual. New Zealand: Department of Civil Engineering, University of Canterbury; 2006.

[58] Naeim F. The seismic design handbook. 2nd ed. Boston: Kluwer Academic; 2011.

[59] Gunturi SKV, Shah HC. In: Building specific damage estimation. Madrid. Rotterdam: Balkema:; 1992. p. 6001–6.

[60] Masi A, Vona M, Mucciarelli M. Selection of natural and synthetic accelerograms for seismic vulnerability studies on reinforced concrete frames. J Struct Eng 2011; 137:367–78.

[61] Kramer SL. Geotechnical earthquake engineering. Prentice-Hall; 1996.

[62] SeismoSoft. SeismoSignal v.5.1.0; 2014: www.seismosoft.com.

[63] Kotsiantis S, Kanellopoulos D, Pintelas P. Data preprocessing for supervised leaning. Int J Comput Sci 2006;1(1):111–7.

[64] Fan C, Chen M, Wang X, Wang J, Huang B. A review on data preprocessing techniques toward efficient and reliable knowledge discovery from building operational data. Front Energy Res 2021;9.

[65] Cortes C, Vapnik V. Support-vector networks. Mach Learn 1995;20(3):273–97.

[66] Breiman L. Random Forests. Mach Learn 2001;45:5–32.

[67] L. Prokhorenkova, G. Gusev, A. Vorobev, A.V. Dorogush, A. Gulin, "CatBoost: unbiased boosting with categorical features", https://arxiv.org/abs/1706.09516.

[68] Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, et al. LightGBM: A highly efficient gradient boosting decision tree. Adv Neural Inf Proces Syst 2017;30:3146–54.

[69] Elith J, Leathwick JR, Hastie T. A working guide to boosted regression trees. J Anim Ecol 2008;77:802–13. https://doi.org/10.1111/j.1365-2656.2008.01390.x.

[70] Geurts P, Ernst D, Wehenkel L. Extremely randomized trees. Mach Learn 2006;63 (1):3–42. ISSN 1573–0565.

[71] Kamiński B, Jakubczyk M, Szufel P. A framework for sensitivity analysis of decision trees. CEJOR 2018;26:135–59. https://doi.org/10.1007/s10100-017-0479-6.

[72] Bishop CM. Pattern Recognition and Machine Learning. Springer; 2006.

[73] Bremner D, Demaine E, Erickson J, Iacono J, Langerman S, Morin P, et al. Output-sensitive algorithms for computing nearest-neighbour decision boundaries. Discret Comput Geom 2005;33:593–604. https://doi.org/10.1007/s00454-004-1152-0.

[74] Duda RO, Hart PE, Stork DH. Pattern Classification. Wiley; 2000.

[75] Conniffe, D., Joan Stone. "A Critical View of Ridge Regression" Journal of the Royal Statistical Society. Series D (The Statistician), 22(3), [Royal Statistical Society, Wiley], 181-87, 1973, https://doi.org/10.2307/2986767.

[76] Tharwat A. Linear vs. quadratic discriminant analysis classifier: a tutorial. Int J Appl Pattern Recogn 2016;3:145–80. https://doi.org/10.1504/IJAPR.2016.079050. ISSN 2049-887X.

[77] S. Theodoridis, K. Koutroumbas, Pattern Recognition, 4th edition, Elsevier, 2008. DOI: 10.1109/TNN.2008.929642.

[78] Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd Edition. Springer Science & Business Media; 2009.

[79] Kégl, B., "The return of ADABOOST.MH: Multi-class Hamming trees", arXiv, 2014, arXiv:1312.6086.

[80] Cramer JS. The early origins of the logit model. Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences 2004;35(4):613–26.

[81] Hossin M, Sulaiman MN. A review on evaluation metrics for data classification evaluations'. Int J Data Mining Knowl Manage Process 2015;5(2):1–11. https://doi.org/10.5121/ijdkp.2015.5201.

[82] Liu Y, Zhou Y, Wen S, Tang C. A strategy on selecting performance metrics for classifier evaluation. International Journal of Mobile Computing and Multimedia Communications 2014;6(4):20–35.

[83] W. Pannakkong, K. Thiwa-Anont, K. Singthong, P. Parthanadee, and J. Buddhakulsomsiri, "Hyperparameter Tuning of Machine Learning Algorithms Using Response Surface Methodology: A Case Study of ANN, SVM, and DBN", Math. Probl. Eng., vol. 2022, Article ID 513719, 2022, doi: 10.1155/2022/8513719.

[84] Kotthoff L, Thornton C, Hoos HH, Hutter F, Leyton-Brown K. Auto-WEKA: Automatic Model Selection and Hyperparameter Optimization in WEKA. In: Hutter F, Kotthoff L, Vanschoren J, editors. Automated Machine Learning: Methods, Systems, Challenges. Cham: Springer; 2019. p. 81–95.

[85] Yang L, Shami A. On hyperparameter optimization of machine learning algorithms: theory and practice. Neurocomputing 2020;415:295–316. https://doi.org/10.1016/j.neucom.2020.07.061.

[86] Probst P, Bischl B, Boulesteix AL. Tunability: importance of hyperparameters of machine learning algorithms", arXiv, arXiv:1802.09596; 2018. doi: 10.48550/arXiv.1802.09596.