Efficient Seismic fragility curve estimation by Active Learning on Support Vector Machines

Rémi Sainct, Cyril Feau, Jean-Marc Martinez, Josselin Garnier

Received 1 January 2000; | Revised 2 January 2000; | Accepted 3 January 2000

DOI: xxx/xxxx

ARTICLE TYPE

Efficient Seismic fragility curve estimation by Active Learning on Support Vector Machines

Rémi Sainct[1] | Cyril Feau*[1] | Jean-Marc Martinez[2] | Josselin Garnier[3]

[1]Den-Service d'études mécaniques et thermiques (SEMT), CEA, Université Paris-Saclay, F-91191 Gif-sur-Yvette, France

[2]Den-Service de thermo-hydraulique et de mécanique des fluides (STMF), CEA, Université Paris-Saclay, F-91191 Gif-sur-Yvette, France

[3]Ecole Polytechnique, CMAP, 91128 Palaiseau Cedex, France

Correspondence
*Email: cyril.feau@cea.fr

Summary

Fragility curves which express the failure probability of a structure, or critical components, as function of a loading intensity measure are nowadays widely used (i) in Seismic Probabilistic Risk Assessment studies, (ii) to evaluate impact of construction details on the structural performance of installations under seismic excitations or under other loading sources such as wind. To avoid the use of parametric models such as lognormal model to estimate fragility curves from a reduced number of numerical calculations, a methodology based on Support Vector Machines coupled with an active learning algorithm is proposed in this paper. In practice, input excitation is reduced to some relevant parameters and, given these parameters, SVMs are used for a binary classification of the structural responses relative to a limit threshold of exceedance. Since the output is not only binary, this is a score, a probabilistic interpretation of the output is exploited to estimate very efficiently fragility curves as score functions or as functions of classical seismic intensity measures.

KEYWORDS:
Fragility curve, Active Learning, Support Vector Machines, seismic intensity measure indicator

1 | INTRODUCTION

In the Seismic Probabilistic Risk Assessment (SPRA) studies performed on industrial facilities, a key point is the evaluation of fragility curves which express the failure probability of a structure, or critical components, as a function of a seismic intensity measure such as peak ground acceleration (PGA) or spectral acceleration. It should be noted that apart from the use in a SPRA framework, fragility curves can be used for making decisions regarding the choice of construction details, to improve the structural performance of installations under seismic excitations (see e.g. [1,2,3,4]). They can also be used to evaluate impact of ground motion characteristics (near-fault type like, broadband, etc.) on the structural vulnerability (see e.g. [4,5]). Finally, it is worth noting that the use of fragility curves is not limited to seismic excitation, they can also be applied to other loading sources such as wind for example [6].

In theory, for complex structures, fragility curves have to be evaluated empirically based on a large number of mechanical analyses requiring, in most cases, nonlinear time-history calculations including both the uncertainties inherent to the system capacity and to the seismic demand, respectively called epistemic and aleatory uncertainties [7]. Nevertheless, the prohibitive computational cost induced by most of nonlinear mechanical models requires the development of numerically efficient methods to evaluate such curves from a minimal number of computations, in particular in industrial contexts.

Following the idea proposed in the early 1980's in the framework of nuclear safety assessment [8], the lognormal parametric model has been widely used in many applications to estimate fragility curves from a reduced number of numerical calculations

(see e.g. [1,2,3,5]). Different methods can be used to determine the parameters of the lognormal model (see e.g. [9,10]), however, the question of the validity of this assumption arises. Typically, in [11] authors show that for a given structure the accuracy of the lognormal curves depends on the ground motion intensity measure, the failure criterion and the employed method for fitting the model. As shown in [12], the class of structures considered may also have an influence on the adequacy of the lognormal model.

The question of the representativity is inevitable with the use of parametric models since, for the complex cases of interest, it is very difficult to verify their validity. To bypass this problem, the need of a numerically efficient non-parametric-based methodology (which would be accurate with a minimum number of mechanical analyses) is necessary. A way to achieve this goal consists in building a metamodel (i.e. a surrogate model of the mechanical analysis) which expresses the statistical relation between seismic inputs and structural outputs. Various metamodeling strategies have been proposed recently in the literature based on, for example, response surfaces ([13,14]), kriging [4] and Artificial Neural Networks (ANNs) [15].

The goal of this paper is twofold. First, it is to propose a simple and efficient methodology for estimating non-parametric fragility curves that allows to reduce the cost of mechanical numerical computations by optimizing their selection. Second, it is to adress the question of the best seismic intensity measure indicator that can to be used as abscissa of the fragilty curves and not be limited to the PGA. To this end, the strategy proposed is based on the use of Support Vector Machines (SVMs) coupled with an active learning algorithm. Thus, input excitation is reduced to some relevant parameters and, given these parameters, SVMs are used for a binary classification of the structural responses relative to a limit threshold of exceedance. It is worth noting that their output is not only binary, it is a "score" that can have a probabilistic interpretation as we will see.

In contrast to classical learning (passive learning), the active learner selects the most useful numerical experiments to be added to the learning data set. The "learners" choose the best instances from a given large set of unlabeled examples. So, the main question in active learning is how to choose new numerical experiments to be labeled. Various methods proposed in active learning by ANNs are presented in [16]. Most are based on the learning of several "learners" ([17,18]). With SVMs, active learning can be done very easily by using only one learner because the distance to the separator hyperplane is a "natural" criterion for selecting new points to "label" [19]. A similar technique using logistic output neural networks can be used by analyzing the logit of the output. But in this case, given the non-linearity of the ANNs, the different learnings of the learner may present a strong variability on the decision boundary.

Finally, as the SVM output is a score, this score can be used as abscissa of the fragility curves. Indeed, a perfect classifier, if it exists, would lead to a fragility curve in the form of a unit step function, i.e. corresponding to a fragility curve "without uncertainty". Nevertheless although certainly not perfect, in the linear binary classification this score can particularly be relevant for engineering purpose since it is a simple linear combination of the input parameters.

To illustrate the proposed methodology, inputs parameters are defined from a set of real accelerograms which is enriched by synthetic accelerograms using the procedure defined in [20], which is based on a parameterized stochastic model of modulated and filtered white-noise process. A brief summary of this model is presented in section 2 of this paper. Moreover a simple inelastic oscillator is considered in order to validate the methodology at a large scale within a Monte Carlo-based approach that does not require any assumption to estimate probabilities of interest. This physical model is presented in section 3. Section 4 is devoted to the presentation of the different classification methods, and the active learning methodology. Section 5 shows how the proposed methodology makes it possible to estimate fragility curves, using either the score functions or classical intensity measure indicators such as the PGA. Finally, a conclusion is presented in section 6.

2 | MODEL OF EARTHQUAKE GROUND MOTION

2.1 | Formulation of the model

Following Rezaeian [21], a seismic ground motion $s(t)$ with $t \in [0, T]$ is modeled as:

$$s(t) = q(t, \boldsymbol{\alpha}) \left[\frac{1}{\sigma_f(t)} \int_{-\infty}^{t} h[t - \tau, \boldsymbol{\lambda}(\tau)] w(\tau) d\tau \right], \tag{1}$$

where $q(t, \boldsymbol{\alpha})$ is a deterministic, non-negative modulating function with a set of parameters $\boldsymbol{\alpha}$, and the process inside the squared brackets is a filtered white-noise process of unit variance: $w(t)$ is a white-noice process, $h(t, \boldsymbol{\lambda})$ denotes the impulse response function (IRF) of the linear filter with a set of parameters $\boldsymbol{\lambda}$, and $\sigma_f(t) = \sqrt{\int_{-\infty}^{t} h^2(t - \tau, \boldsymbol{\lambda}(\tau)) d\tau}$ is the standard deviation of the process defined by the integral in equation 1.

In order to achieve spectral nonstationarity of the ground motion, the parameters $\boldsymbol{\lambda}$ of the filter depend on the time τ of application of the pulse; thus the standard deviation σ depend on t. Still following Rezaeian, we choose for the impulse response function:

$$\begin{aligned} h[t-\tau, \boldsymbol{\lambda}(\tau)] &= \frac{\omega_f(\tau)}{\sqrt{1-\zeta_f^2}} \exp\left[-\zeta_f \omega_f(\tau)(t-\tau)\right] \sin\left[\omega_f(\tau)\sqrt{1-\zeta_f^2}(t-\tau)\right] \quad \text{if } t \geq \tau, \\ &= 0 \quad \text{otherwise,} \end{aligned} \tag{2}$$

where $\boldsymbol{\lambda}(\tau) = [\omega_f(\tau), \zeta_f]$ is the set of parameters, $\omega_f(\tau)$ is the natural frequency (dependent on the time of application of the pulse) and $\zeta_f \in [0, 1]$ is the (constant) damping ratio. A linear form is chosen for the frequency: $\omega_f(\tau) = \omega_0 + \frac{\tau}{T}(\omega_n - \omega_0)$. The modulating function $q(t, \boldsymbol{\alpha})$ is defined piecewisely:

$$\begin{aligned} q(t, \boldsymbol{\alpha}) &= 0 && \text{if } \ t \leq T_0, \\ &= \alpha_1 \left(\tfrac{t-T_0}{T_1-T_0}\right)^2 && \text{if } \ T_0 \leq t \leq T_1, \\ &= \alpha_1 && \text{if } \ T_1 \leq t \leq T_2, \\ &= \alpha_1 \exp\left[-\alpha_2 (t-T_2)^{\alpha_3}\right] && \text{if } \ t \geq T_2. \end{aligned} \tag{3}$$

The modulation parameters are thus: $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3, T_0, T_1, T_2)$. The initial delay T_0 is used in parameter identification for real ground motions, but it is not used in simulations (we choose $T_0 = 0$). To summarize, the generated signals are associated with 8 real parameters: $(\alpha_1, \alpha_2, \alpha_3, T_1, T_2, \omega_0, \omega_n, \zeta_f)$. Finally, a high-pass filter is used as post-processing to guarantee zero residuals in the acceleration, velocity and displacement. The corrected signal $\ddot{u}(t)$ is the solution of the differential equation:

$$\ddot{u}(t) + 2\omega_c \dot{u}(t) + \omega_c^2 u(t) = s(t), \tag{4}$$

where $\omega_c = 0.2$ Hz is the corner frequency. Due to high damping of the oscillator (damping ratio of 100%), it is clear that $u(t)$, $\dot{u}(t)$ and $\ddot{u}(t)$ all vanish shortly after the input process $s(t)$ has vanished, thus assuring zero residuals for the simulated ground motion. In the rest of this paper we will use $s(t)$ for the corrected signal $\ddot{u}(t)$.

2.2 | Parameter identification

The first step to generate artificial signals is to identify the 9 model parameters for every real signal $a(t)$ from our given database of $N_r = 97$ acceleration records (selected from the European Strong Motion Database[22] for $5.5 < M < 6.5$ and $R < 20$km, where M is the magnitude and R the distance from the epicenter). Following Rezaeian[21,20], the modulation parameters $\boldsymbol{\alpha}$ and the filter parameters $\boldsymbol{\lambda}$ are identified independently as follows.

2.2.1 | Modulation parameters

For a target recorded accelerogram $a(t)$, we determine the modulation parameters $\boldsymbol{\alpha}$ by matching the cumulative energy of the accelerogram $E_a(t) = \int_0^t a^2(\tau) d\tau$ with the expected cumulative energy $E_s(t)$ of the stochastic process $s(t)$, which does not depend on the filter parameters (if the high-pass postprocessing is neglected):

$$\begin{aligned} E_s(t) &= \mathbb{E}\left[\int_0^t s^2(\tau) d\tau\right] = \mathbb{E}\left[\int_0^t q^2(\tau, \boldsymbol{\alpha}) \tfrac{1}{\sigma_f^2(\tau)} \left(\int_{-\infty}^{\tau} h[\tau - u, \boldsymbol{\lambda}(u)] w(u) du\right)^2 d\tau\right] \\ &= \int_0^t q^2(\tau, \boldsymbol{\alpha}) \tfrac{1}{\sigma_f^2(\tau)} \mathbb{E}\left[\left(\int_{-\infty}^{\tau} h[\tau - u, \boldsymbol{\lambda}(u)] w(u) du\right)^2\right] d\tau = \int_0^t q^2(\tau, \boldsymbol{\alpha}) d\tau. \end{aligned} \tag{5}$$

Thanks to the definition of $\sigma_f(t)$, this expected energy only depends on the modulation parameters $\boldsymbol{\alpha}$. To match the two cumulative energy terms, we minimize the integrated squared difference between them:

$$\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \int_0^T \left[E_s(t) - E_a(t)\right]^2 dt = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \int_0^T \left[\int_0^t q^2(\tau, \boldsymbol{\alpha}) d\tau - \int_0^t a^2(\tau) d\tau\right]^2 dt. \tag{6}$$

This minimization is done with the Matlab function fminunc. Note that the PGA of the generated signal is not necessarily equal to that of the recorded one on average.

2.2.2 | Filter parameters

For the filter parameters $\boldsymbol{\lambda} = (\omega_0, \omega_n, \zeta_f)$, we use the zero-level up-crossings, and the positive minima and negative maxima of the simulated signal $s(t)$ and target signal $a(t)$. These quantities do not depend on scaling, thus we use only the un-modulated process

$$y(t) = \int_{-\infty}^{t} \frac{h[t-\tau, \boldsymbol{\lambda}(\tau)]}{\sigma_f(t)} w(\tau) d\tau. \tag{7}$$

For a given damping ratio ζ_f, we can identify the frequencies (ω_0, ω_n) by matching the cumulative count $N_a(t)$ of zero-level up-crossings of the target signal $a(t)$ with the same expected cumulative count $N_x(t)$ for the simulated signal, given by:

$$N_x(t) = \int_{0}^{t} \nu(\tau) r(\tau) d\tau, \tag{8}$$

where $\nu(\tau)$ is the mean zero-level up-crossing rate of the process $y(t)$ and $r(\tau)$ is an adjustment factor due to discretization (usually between 0.75 and 1). Since $y(t)$ is a Gaussian process with zero mean and unit variance, the mean rate $\nu(\tau)$, after simplification, is given by:

$$\nu(t) = \frac{\sigma_{\dot{y}}(t)}{2\pi}, \tag{9}$$

where $\sigma_{\dot{y}}(t)$ is the standard deviation of the time derivative $\dot{y}(t)$ of the process:

$$\sigma_{\dot{y}}(t)^2 = \int_{-\infty}^{t} \left[\dot{h}(t-\tau, \boldsymbol{\lambda}(\tau)) - h(t-\tau, \boldsymbol{\lambda}(\tau)) \frac{\int_{-\infty}^{t} h(t-u, \boldsymbol{\lambda}(u)) \dot{h}(t-u, \boldsymbol{\lambda}(u)) du}{\sigma_f(t)^2} \right]^2 \frac{d\tau}{\sigma_f(t)^2}, \tag{10}$$

assuming we neglect integrals over a fraction of a time step in the discretization.

To identify the damping ratio ζ_f, we use the cumulative count of positive minima and negative maxima. Indeed, in a narrow-band process (ζ_f close to 0), almost all maxima are positive and almost all minima are negative, but the rate increases with increasing bandwidth (larger ζ_f). An explicit formulation exists for this rate but it involves computing the second derivative of $y(t)$[23], thus it is easier to use a simulation approach, by counting and averaging the negative maxima and positive minima in a sample of simulated realizations of the process, then choosing the value that minimizes the difference between real and simulated cumulative counts.

2.3 | Simulation of ground motions

So far, we have defined a model of earthquake ground motions, and explained how to identify each of the parameters from a single real signal $a(t)$. The model then allows one to generate any number of artificial signals, thanks to the white noise $w(t)$. However, these signals would all have very similar features; in order to estimate a fragility curve, we need to be able to generate artificial signals over a whole range of magnitudes, with realistic associated probabilities. Thus, we have to add a second level of randomness in the generation process, coming from the parameters themselves. With Rezaian's method we identified all the model parameters $\theta = (\boldsymbol{\alpha}, \boldsymbol{\lambda})$ for each of the $N_r = 97$ acceleration records, giving us N_r data points in the parameter space ($\mathbb{R}^8$ in this case). Then, to define the parameters' distribution, we use a Gaussian kernel density estimation (KDE) with a multivariate bandwidth estimation, following Kristan[24].

Let $(\theta_1, \theta_2, \dots, \theta_{N_r})$ be a multivariate independent and identically distributed sample drawn from some distribution with an unknown density $p(\theta)$, $\theta \in \mathbb{R}^d$. The kernel density estimator p_{KDE} is:

$$p_{KDE}(\theta) = \frac{1}{N_r} \sum_{1}^{N_r} \phi_{\mathbf{H}}(\theta - \theta_i), \tag{11}$$

where $\phi_{\mathbf{H}}$ is a Gaussian kernel centered at 0 with covariance matrix $\mathbf{H}$. A classical measure of closeness of the estimator $p_{KDE}(\theta)$ to the unknown underlying probability density function (pdf) is the *asymptotic mean integrated squared error* (AMISE), defined as:

$$\mathrm{AMISE}(\mathbf{H}) = (4\pi)^{-d/2}|\mathbf{H}|^{-1/2}N_r^{-1} + \frac{1}{4}d^2 \int \mathrm{Tr}^2\left[\mathbf{H}\,\mathrm{Hess}_p(\theta)\right] d\theta, \tag{12}$$

where Tr is the trace operator and Hess_p is the Hessian of p. If we write $\mathbf{H} = \beta^2\mathbf{F}$ with $\beta \in \mathbb{R}$ and we suppose that $\mathbf{F}$ is known, then the AMISE is minimized for:

$$\beta_{opt} = \left[d(4\pi)^{d/2}N_r R(p,\mathbf{F})\right]^{-\frac{1}{d+4}}, \tag{13}$$

where R still depends on the underlying (and unknown) distribution $p(\theta)$:

$$R(p,\mathbf{F}) = \int \mathrm{Tr}^2\left[\mathbf{F}\,\mathrm{Hess}_p(\theta)\right] d\theta. \tag{14}$$

Following Kristan[24], $\mathbf{F}$ can be approximated by the empirical covariance matrix $\hat{\mathbf{\Sigma}}^{smp}$ of the observed samples and $R(p,\mathbf{F})$ can be approximated by

$$\hat{R} = \left(\frac{4}{(d+2)N_r}\right)^{-\frac{4}{d+4}} \sum_{i,j=1}^{N_r} \phi_{\hat{\mathbf{G}}}(\theta_i - \theta_j)\left(\frac{2}{N_r}(1-2m_{ij}) + (1-m_{ij})^2\right), \tag{15}$$

where

$$m_{ij} = (\theta_i - \theta_j)^T\hat{\mathbf{G}}^{-1}(\theta_i - \theta_j), \qquad \hat{\mathbf{G}} = \left(\frac{4}{(d+2)N_r}\right)^{\frac{2}{d+4}}\hat{\mathbf{\Sigma}}^{smp}. \tag{16}$$

The estimator $p_{KDE}(\theta) = \frac{1}{N_r}\sum_1^{N_r}\phi_{\mathbf{H}}(\theta - \theta_i)$ is now fully defined. The simulation of an artificial ground motion thus requires three steps:

- choose an integer $i \in [\![1, N_r]\!]$ with a uniform distribution;
- sample a vector $\mathbf{y}$ from a Gaussian distribution with pdf $\phi_{\mathbf{H}}$ centered at 0 with covariance matrix $\mathbf{H}$, and let $\theta = \theta_i + \mathbf{y}$;
- sample a white noise $w(\tau)$ and compute the signal using (1), with parameters $(\boldsymbol{\alpha}, \boldsymbol{\lambda}) = \theta$.

For this article we generated $N_s = 10^5$ artificial seismic ground motions $s_i(t)$ using this method.

3 | PHYSICAL MODEL

3.1 | Equations of motion

For the illustrative application of the methodology developed in this paper, a nonlinear single degree of freedom system is considered. Indeed, despite its extreme simplicity, such model may reflect the essential features of the nonlinear responses of some real structures. Moreover, in a probabilistic context requiring Monte Carlo simulations, it makes it possible to have reference results with reasonable numerical cost. Its equation of motion reads:

$$\ddot{z}_i(t) + 2\beta\omega_L\dot{z}_i(t) + f_i^{nl}(t) = -s_i(t), \quad i \in [\![1, N_s]\!] \tag{17}$$

where $\dot{z}_i(t)$ and $\ddot{z}_i(t)$ are respectively the relative velocity and acceleration of the unit mass of the system submitted to the ith artificial seismic ground motion $s_i(t)$ with null initial conditions in velocity and displacement. In equation 17, β is the damping ratio, $\omega_L = 2\pi f_L$ is the circular frequency and $f_i^{nl}(t)$ is the nonlinear resisting force. In this study, $f_L = 5$ Hz, $\beta = 2\%$, the yield limit is $Y = 5.10^{-3}$ m, and the post-yield stiffness, defining kinematic hardening, is equal to 20% of the elastic stiffness. Moreover, we call $\tilde{z}_i(t)$ the relative displacement of the associated linear system, that is assumed to be known in the sequel, whose equation of motion is:

$$\ddot{\tilde{z}}_i(t) + 2\beta\omega_L\dot{\tilde{z}}_i(t) + \omega_L^2\tilde{z}_i(t) = -s_i(t), \tag{18}$$

and we set:

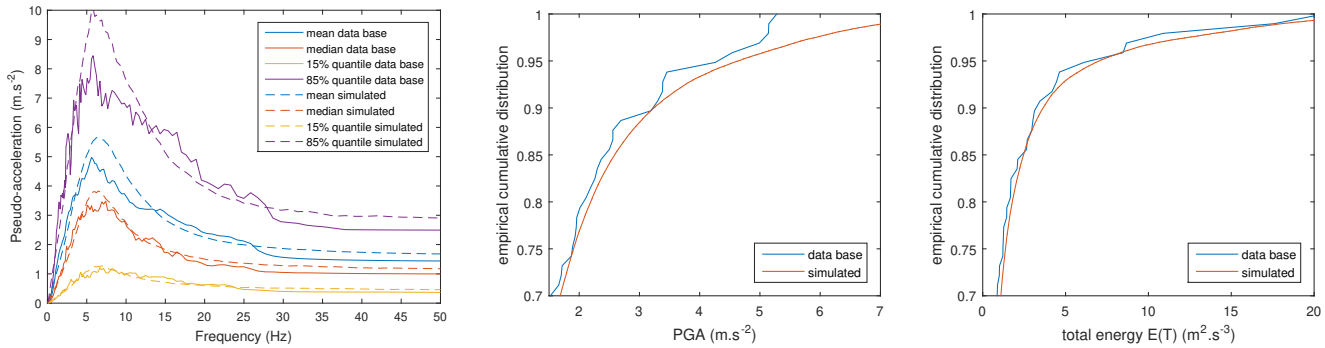
$$Z_i = \max_{t\in[0,T]}|z_i(t)|, \tag{19}$$

$$L_i = \max_{t\in[0,T]} |\tilde{z}_i(t)|. \tag{20}$$

In this work, equations 17 and 18 are solved numerically with a finite-difference method.

3.2 | Response spectrum

Using the linear equation 18, we can compare the response spectra of the $N_r = 97$ recorded accelerograms with that of the $N_s = 10^5$ simulated signals. Figure 1 a shows this comparison for the average spectrum, as well as the 0.15, 0.5 and 0.85 quantiles. It can be seen that the simulated signals have statistically the same response spectra than the real signals, although at high frequency ($f_L > 30$ Hz), the strongest simulated signals have higher responses than the real ones. This may be due to the fact that the model conserves energy (equation 5), while the selection of acceleration records from ESMD is based on magnitude. To illustrate this, figures 1 b and 1 c show the empirical cumulative distribution functions of the PGA and total energy. While there is a good match for the energy, the PGA of the strongest simulated signals is slightly higher than for the real ones.



(a) (b) (c)

FIGURE 1 Comparison between the real and simulated data bases. (a) Response spectra for 2% damping ratio. Zoom on the empirical cumulative distribution functions of (b) the PGA and (c) total energy.

3.3 | Choice of the seismic intensity measures

Let $\mathcal{B} = (s_i(t))_{i\in[\![1,N_s]\!]}$ be our database of simulated ground motions, and $\theta_i = (\boldsymbol{\alpha}_i, \boldsymbol{\lambda}_i) \in \mathbb{R}^8$ the associated modulating and filter parameters. For every signal $s_i(t)$, we also consider:

- the peak ground acceleration $PGA_i = \max_{t\in[0,T]} |s_i(t)|$;
- the maximum velocity (or Peak Ground Velocity) $V_i = \max_{t\in[0,T]} \left|\int_0^t s_i(\tau)d\tau\right|$;
- the maximum displacement (or Peak Ground Displacement) $D_i = \max_{t\in[0,T]} \left|\int_0^t \int_0^\tau s_i(u)du d\tau\right|$;
- the total energy $E_i = E_{s_i}(T) = \int_0^T s_i^2(\tau)d\tau$;
- the maximum linear displacement L_i of the structure (equation 20). Conventionnaly, this is spectral acceleration $(\omega_L^2 L_i)$ which is considered as intensity measure indicator. Nevertheless, since here the variable of interest is a non-linear displacement, it is more suitable to use spectral displacement.

Thus, for each simulated signal we have a vector $X_i^\star = (\boldsymbol{\alpha}_i, \boldsymbol{\lambda}_i, PGA_i, V_i, D_i, E_i, L_i) \in \mathbb{R}^{13}$ of 13 real parameters. We want to know whether the maximum total displacement Z_i of the structure is greater than a certain threshold, for example twice the elasticity limit Y.

4 | BINARY CLASSIFICATION

4.1 | Preprocessing of the training data

The signals whose maximum linear displacement is less than the elasticity limit Y are not interesting, since we know they do not reach the threshold:

$$L_i < Y \quad \Rightarrow \quad Z_i = L_i \quad \text{and thus} \quad Z_i < Y.$$

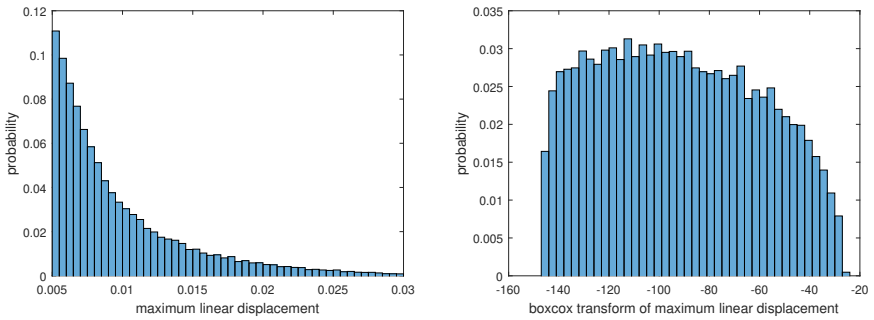
This discarded 66% of the simulated signals. We also discarded a few signals (0.3%) whose maximum linear displacement was too high ($L_i > 6Y$), since the mechanical model we use is not realistic beyond that level. Therefore, we ended with a subset I of our database such that:

$$\forall i \in I \quad L_i \in [Y, 6Y].$$

We have kept $N = 33718$ simulated signals out of a total of $N_s = 10^5$. On those N signals, a Box-Cox transform was applied on each component of $X_i^\star$. This non-linear step is critical for the accuracy of the classification, especially when we later use linear SVM classifiers. The Box-Cox transform is defined by:

$$BC(x, \delta) = \begin{cases} \dfrac{x^\delta - 1}{\delta} & \text{if } \delta \neq 0 \\ \log(x) & \text{if } \delta = 0. \end{cases} \tag{21}$$

The parameter δ is optimized, for each component, assuming a normal distribution and maximizing the log-likelihood. Figure 2 shows that L is heavily modified by this transformation, with an optimal parameter of $\delta = -0.928$.



(a) **(b)**

FIGURE 2 Histograms of L, (a) before and (b) after a Box-Cox transform with parameter $\delta = -0.928$.

Finally, each of the 13 components was standardized, thus forming the training database $\mathcal{X} = \{X_1, \dots, X_N\}$ with $X_i \in \mathbb{R}^{13}$.

4.2 | Simple classifiers

At the most basic level, a binary classifier is a labeling function

$$\begin{aligned} \hat{l} : \mathbb{R}^d &\longrightarrow \{-1; 1\} \\ X &\longmapsto \hat{l}(X), \end{aligned} \tag{22}$$

that, given a vector $X \in \mathbb{R}^d$ corresponding to a seismic signal $s(t)$, gives us an estimated label $\hat{l}$. In our setting, the true label l_i of instance X_i is 1 if the displacement Z_i is greater than the damage threshold $2Y$, and -1 otherwise:

$$l_i = \mathrm{sgn}(\mathrm{Z_i} - 2\mathrm{Y}) = \begin{cases} 1 & \text{if} \quad Z_i > 2Y, \\ -1 & \text{otherwise.} \end{cases} \tag{23}$$

Note that the true label l_i is not in general a function of the vector X_i, since it depends on the full signal $s_i(t)$ when X_i only gives us macroscopic measures of the signal; therefore, a perfect classifier $\hat{l}(X_i)$ may not exist.

One of the simplest choice for a classifier is to look at only one component of the vector X. For example, it is obvious that the PGA is highly correlated with the maximum total displacement Z_i; therefore, we can define the PGA classifier $\hat{l}_{PGA}$ as:

$$\hat{l}_{PGA}(X) = \text{sgn}(\text{PGA} - \text{M}) \tag{24}$$

where M is a given threshold. Moving the threshold up results in less false positives ($\hat{l} = 1$ when the real label is $l = -1$) but more false negatives ($\hat{l} = -1$ when the real label is $l = 1$); and moving the threshold down results in the opposite. Therefore, there exists a choice of M such that the number of false positives and false negatives are equal, as can be seen in figure 3 .

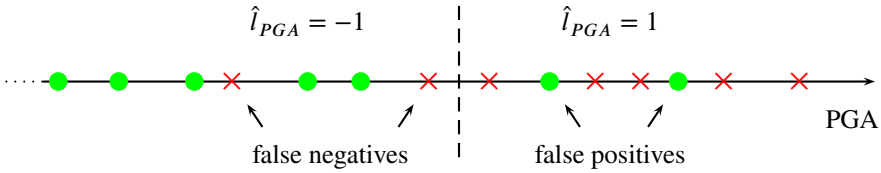


FIGURE 3 Choice of the threshold for the binary PGA classifier $\hat{l}_{PGA}$.

Note that this choice does not guarantee that the *total* number of misclassifications is minimal. Similarly, we can also define a classifier $\hat{l}_L$ based on the maximum linear displacement L, since the linear displacement is also highly correlated with the total displacement. These two simple classifiers give us a baseline to measure the performance of more advanced classifiers.

4.3 | SVMs and active learning

4.3.1 | Support vector machines

In machine learning, support vector machines (SVMs) are supervised learning models used for classification and regression analysis. In the linear binary classification setting, given a training data set $\{X_1, \dots, X_n\}$ that are vectors in $\mathbb{R}^d$, and their labels $\{l_1, \dots, l_n\}$ in $\{-1, 1\}$, the SVM is a hyperplane of $\mathbb{R}^d$ that separates the data by a maximal margin. More generally, SVMs allow one to project the original training data set $\{X_1, \dots, X_n\}$ onto a higher dimensional feature space via a Mercer kernel operator K. The classifier then associates to each new signal X a score $f_n(X)$ given by:

$$f_n(X) = \sum_{i=1}^{n} \alpha_i K(X_i, X). \tag{25}$$

A new seismic signal represented by the vector X has an estimated label $\hat{l}$ of 1 if $f_n(X) > 0$, -1 otherwise. In a general SVM setting, most of the labeled instances X_i have an associated coefficient α_i equal to 0; the few vectors X_i such that $\alpha_i \neq 0$ are called "support vectors", thus the name "support vector machine". This historical distinction among labeled instances is less relevant in the case of active learning (see next section), since most of the α_i are non-zero. In the linear case, $K(X_i, X)$ is just the scalar product in $\mathbb{R}^d$, and the score is:

$$f_n(X) = W^T X + c, \tag{26}$$

where $W \in \mathbb{R}^d$ and $c \in \mathbb{R}$ depend on the coefficients α_i. Another commonly used kernel is the radial basis function kernel (or RBF kernel) $K(U, V) = e^{-\gamma(U-V)\cdot(U-V)}$, which induces boundaries by placing weighted Gaussians upon key training instances.

4.3.2 | Active learning

Computing the total displacement Z_i of the structure (and thus the label l_i) is very costly for a complex structure, limiting the size of the training data. Fortunately, it is possible to make accurate classifiers using only a limited number of labeled training instances, using active learning.

In the case of pool-based active learning, we have, in addition to the labeled set $\mathcal{L} = \{X_1, \dots, X_n\}$, access to a set of unlabeled samples $\mathcal{U} = \{X_{n+1}, \dots, X_N\}$ (therefore we have $\mathcal{X} = \mathcal{L} \cup \mathcal{U}$). We assume that there exists a way to provide us with a label for any sample X_i from this set (in our case, running a full simulation of the physical model using signal $s_i(t)$), but the labeling cost is high. After labeling a sample, we simply add it to our training set. In order to improve a classifier it seems intuitive to

query labels for samples that cannot be easily classified. Various querying methods are possible[25,19], but the method we present here only requires to compute the score $f_n(X)$ for all samples in the unlabeled set, then to identify a sample that reaches the minimum of the absolute value $|f_n(X)|$, since a score close to 0 means a high uncertainty for this sample. Thus, we start with $n = 2$ samples j_1 and j_2, labeled $+1$ and -1. Recursively, if we know the labels of signals $j_1, \dots, j_n$:

- we compute the SVM classifier associated with the labeled set $\{(X_{j_1}, l_{j_1}), \dots, (X_{j_n}, l_{j_n})\}$;
- for each unlabeled instance X_i, $i \in [\![1, N]\!] \backslash \{j_1, \dots, j_n\}$, we compute its score $f_n(X_i) = \sum_{k=1}^{n} \alpha_k K(X_{j_k}, X_i)$;
- we query the instance with maximum uncertainty for this classifier:

$$j_{n+1} = \underset{i \in [\![1,N]\!] \backslash \{j_1,\dots,j_n\}}{\operatorname{argmin}} |f_n(X_i)|, \tag{27}$$

  and compute the corresponding maximum total displacement $Z_{j_{n+1}}$ by running a full simulation of the physical model;
- the instance $(X_{j_{n+1}}, l_{j_{n+1}} = \mathrm{sgn}(\mathrm{Z_{j_{n+1}}} - 2\mathrm{Y}))$ is added to the labeled set.

4.3.3 | Choice of the starting points

The active learner needs two starting points, one on each side of the threshold. After the preprocessing step, about 17% of all remaining instances have a displacement greater than the threshold (although this precise value is usually unknown). It can be tempting to choose, for example, the signal with the smallest PGA as j_1 and the signal with the biggest PGA as j_2. However, running simulations with these signals is costly and give us a relatively useless information. We prefer to choose the starting points randomly, which also allows us to see how this randomness affects the final performance of the classifier.

The linear displacement L_i and the PGA_i of a signal are both obviously strongly correlated with the displacement Z_i. As a consequence, it is preferable that the starting points respect the order for these two variables:

$$Z_{j_1} < 2Y < Z_{j_2}, \quad L_{j_1} < L_{j_2} \quad \text{and} \quad PGA_{j_1} < PGA_{j_2}. \tag{28}$$

Indeed, if j_1 and j_2 are such that, for example, $Z_{j_1} < 2Y < Z_{j_2}$ but $PGA_{j_1} > PGA_{j_2}$, then the active learner starts by assuming that the PGA and displacement have a negative correlation, and it can take many iterations before it "flips"; in some rare instances the classifier performs extremely poorly for several hundreds of iterations. Thus, the starting points j_1 and j_2 are chosen such that equation (28) is automatically true, using quantiles of the PGA and linear displacement. j_1 is chosen randomly among the instances whose PGA is smaller than the median PGA and whose linear displacement is smaller than the median linear displacement:

$$j_1 \in \left\{ i \in [\![1, N]\!] \quad | \quad PGA_i < D_5(PGA) \quad \& \quad L_i < D_5(L) \right\}, \tag{29}$$

where $D_5(X)$ denotes the median of set X. It is almost certain that any instance in this set satisfies $Z_i < 2Y$ and thus $l_i = -1$. Similarly, j_2 is chosen using the 9th decile of both PGA and linear displacement:

$$j_2 \in \left\{ i \in [\![1, N]\!] \quad | \quad PGA_i > D_9(PGA) \quad \& \quad L_i > D_9(L) \right\}, \tag{30}$$

where $D_9(X)$ denotes the 9th decile of a set X. The probability that $Z_i > 2Y$ in this case was found to be 97%. If we get unlucky and $Z_i < 2Y$ then we discard this signal and choose another one.

4.4 | ROC curve and precision/recall breakeven point

The SVM classifier gives an estimated label $\hat{l}_i$ to each signal s_i depending on its score $\hat{l}_i = \mathrm{sgn}(\mathrm{f_n(X_i)})$. As for the simple classifiers (section 4.2), we can set a non-zero limit β, and define the classifier as:

$$\hat{l}_i(\beta) = \mathrm{sgn}(\mathrm{f_n(X_i)} - \beta), \quad \beta \in \mathbb{R} \tag{31}$$

If $\beta > 0$, then the number of false positives ($l_i = -1$ and $\hat{l}_i = 1$) is smaller, but the number of false negatives ($l_i = 1$ and $\hat{l}_i = -1$) is bigger, relative to the $\beta = 0$ case, and the opposite is true if we choose $\beta < 0$. Taking all possible values for $\beta \in \mathbb{R}$

defines the *receiver operating characteristic* curve, or ROC curve. The area under the ROC curve is a common measure for the quality of a binary classifier. The classifier is perfect if there exists a value of β such that all estimated labels are equal to the true labels; in this case the area under the curve is equal to 1. Figure 4 shows one example of active learning, with ROC curves corresponding to different numbers of labeled signals. As expected, the classifier improves on average when the labeled set gets bigger; and the active learner becomes better than the simple PGA classifier as soon as $n \geq 10$.

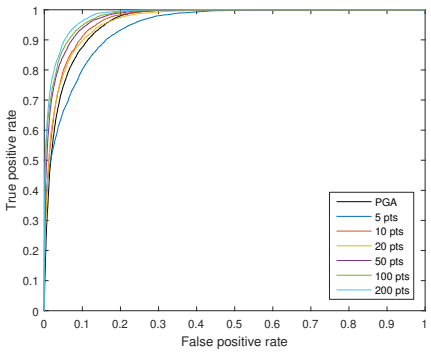


FIGURE 4 ROC curves for the PGA classifier (black) and for 6 active learners after n iterations ($n = 5, 10, 20, 50, 100$ and 200).

Another metric can be used to measure performance: the precision/recall breakeven point[25]. Precision is the percentage of samples a classifier labels as positive that are really positive. Recall is the percentage of positive samples that are labeled as positive by the classifier. By altering the decision threshold on the SVM we can trade precision for recall, until both are equal, therefore defining the precision/recall breakeven point. In this case the number of false positives and false negatives are equal (see figure 3). This value is very easy to obtain from a practical point of view. Let us denote by N_+ the number of instances where the displacement is greater than the threshold (on a total of N signals in the database):

$$N_+ = \#\left\{i \in [\![1, N]\!] \quad | \quad l_i = 1\right\}. \tag{32}$$

We sort all instances according to their score, i.e. we find a permutation σ such that:

$$f_n(X_{\sigma(1)}) \leq \cdots \leq f_n(X_{\sigma(N)}). \tag{33}$$

Then the precision/recall breakeven point (PRBP) is equal to the proportion of positive instances among the N_+ instances with the highest score:

$$\text{PRBP} = \frac{\#\left\{i \in [\![1, N]\!] \quad | \quad l_i = 1 \quad \& \quad \sigma(i) > N - N_+\right\}}{\#\left\{i \in [\![1, N]\!] \quad | \quad \sigma(i) > N - N_+\right\}} = \frac{\#\left\{i \in [\![1, N]\!] \quad | \quad l_i = 1 \quad \& \quad \sigma(i) > N - N_+\right\}}{N_+} \tag{34}$$

This criteria does not depend on the number of true negatives (unlike the false positive rate, used in the ROC curve). In particular, it is not affected by our choice of preprocessing of the training data, where we discarded all the weak signals ($L_i < Y$). (both metrics are affected by our choice to discard the very strong signals ($L_i > 6Y$), but the effect is negligible in both cases).

4.5 | Results

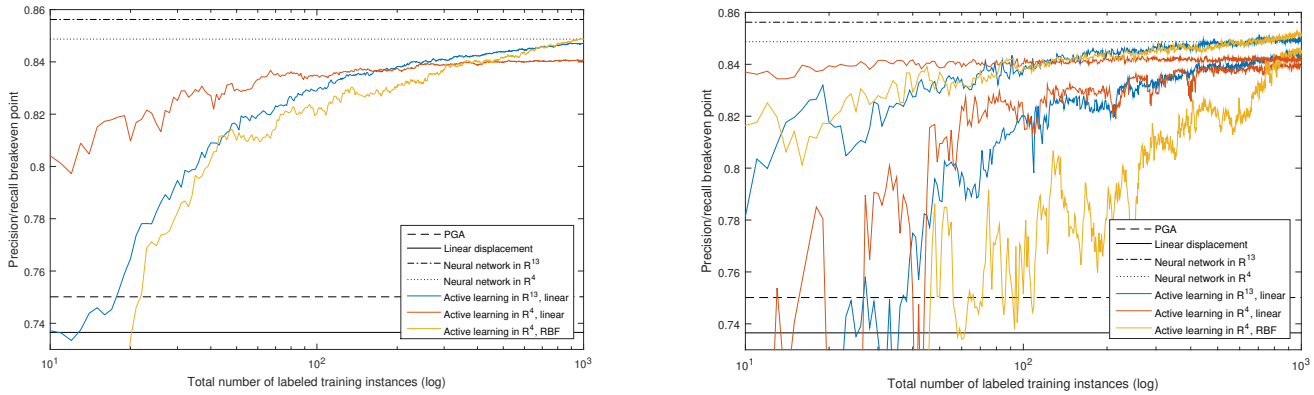
We now compare different classifiers with the precision/recall breakeven point. More precisely, we compare different *orderings* of all signals, since only the order matters to the PRBP; for instance, the PGA does not give directly a label, but we can compute the PRBP of the PGA classifier with equation 34 using the permutation σ_{PGA} that sorts the PGA of all signals. We can thus compare:

1. the simple PGA and maximum linear displacement classifiers $\hat{l}_{PGA}$ and $\hat{l}_L$, defined in section 4.2;

2. neural networks, trained with all instances and all labels (ie, with the $N = 33718$ signals and labels), with either all 13 parameters, or just 4 of them: (L, PGA, V, ω_0) (the linear displacement, peak ground acceleration, peak ground velocity and filter frequency, see section 4.7 for justification of this choice);

3. SVMs given by our active learning methods.

The neural networks we used are full-connected Multi Layered Perceptrons (MLPs) with 2 layers of 26 and 40 neurons for $X \in \mathbb{R}^4$, and two layers of 50 and 64 neurons for $X \in \mathbb{R}^{13}$. In the active learning category, the performance depends on the number of iterations (between 10 and 1000). So, the results shown in figure 5 are functions of the number n of labeled training instances, in logarithmic scale. The simple classifiers and the neural networks are represented as horizontal lines, since they do not depend on n. Moreover, this figure shows the PRBP of (i) a linear SVM using all 13 parameters (in blue), (ii) a linear SVM using only 4 parameters (L, PGA, V, ω_0) (in red) and (iii) a radial basis function (RBF) SVM, using the same 4 parameters (in yellow). As active learners depend on the choice of the first two samples, results of figure 5 a are obtained choosing 20 pairs of starting points (j_1, j_2), then averaging the performance, knowing that the same starting points were used for all three types of SVMs. For completeness, figure 5 b shows the worst and best performances of the 3 classifiers on the 20 test cases.



(a) **(b)**

FIGURE 5 Performances of 3 active learning classifiers over 20 test cases. (a) Average. (b) Worst and best.

Figure 5 a shows that active learning gives a much better classifier than the standard practice of using a single parameter (usually the PGA). The linear SVM with only 4 variables has initially the best performance on average, up to 150/200 iterations. The full linear SVM with 13 variables is better when the number of iterations is at least 200. The RBF kernel in $\mathbb{R}^4$ appears to have the best performance with 1000 labeled instances, outperforming the neural network in $\mathbb{R}^4$; however, it has a higher variability, as can be seen in figure 5 b. Radial basis function SVMs with 13 parameters seem to always perform very poorly, and are not represented here. Figure 5 b shows the lowest and highest score of all 20 test cases, independently for each number of iterations (one active learner can perform poorly at some point, and much better later, or the other way around). So, in conclusion, (i) active learners need a minimum of 30-40 iterations, otherwise they can end up worse than using the simpler PGA classifier, (ii) between 50 and 200 iterations, the linear SVM in $\mathbb{R}^4$ is the best choice, and has a relatively small variability and (iii) the RBF kernel seems quite unpredictable for less than 1000 iterations, and its performance depends wildly on the starting points, probably because of over-fitting.

4.6 | Results for different settings

Our methodology is very general and can be applied to a variety of structures. As an example, we compared the same classifiers on two structures with two different main frequencies, 2.5 Hz and 10 Hz, instead of the original 5 Hz. The elasticity limit Y was also changed so that approximately one third of all signals result in inelastic displacement: $Y = 9 \cdot 10^{-3}$ m for the 2.5 Hz structure, $Y = 5 \cdot 10^{-3}$ m for 5 Hz and $Y = 1 \cdot 10^{-3}$ m for 10 Hz. The failure threshold was always chosen as $2Y$, which resulted

in about 8.8% of all signals attaining it for the 2.5 and 10 Hz cases, compared to 5.7% in the 5 Hz setting. As shown in figure 6 , the performances of active learners are very similar to the 5 Hz case, and the same conclusions apply. The performances of classifiers based on a single parameter, on the other hand, can vary a lot depending on the frequence of the structure: the PGA classifier provides a good classifier at high frequency (PRBP= 0.797 at 10 Hz) but performs poorly at low frequency (0.6 at 2.5 Hz, it does not appear in figure 6); while the linear displacement does the opposite (PRBP= 0.798 at 2.5 Hz, but PRBP= 0.69 at 10 Hz). These results show that the active learning methodology is not just more precise, but also more flexible than the simple classifiers, and that with just 50 to 200 simulations it approaches the performance of a neural network using 33718 simulations.

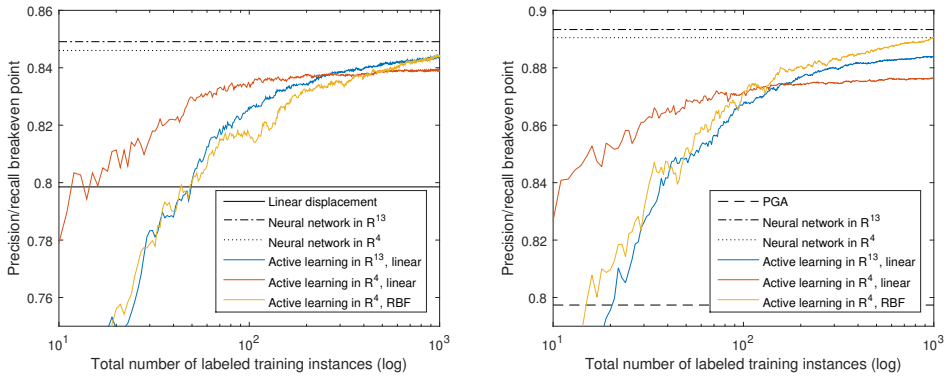


FIGURE 6 Average performance of active learning classifiers at (a) 2.5 Hz and (b) 10 Hz, over 20 test cases.

4.7 | Remark about the dimension reduction

In the linear case the score is equal to the distance to a hyperplane: $f_n(X) = W^T X + c$ (see equation 26). Therefore, we can see which of the components of X are the most important for the classification simply by looking at the values of the components of W. Figure 7 shows that the values of W are roughly the same for all 20 test cases. After 1000 iterations, the coefficients for the PGA and maximum linear displacement L end up between 3 and 4, the value for the maximum velocity V is around 1, and the value for the signal main frequency ω_0 is around -1. The other 9 components of W (when working with $X \in \mathbb{R}^{13}$) are all between -1 and 1, but are smaller (in absolute value) than these 4 components. Note that comparing the values of the components of W is only possible because the components of X are standardized in the first place (cf. the preprocessing in section 4.1). As can be seen in the previous sections, reducing the dimension from 13 to 4 allows for a faster convergence, although the converged classifier is usually less precise. Continuing the active learning after 1000 iterations changes only marginally the results; even with $X \in \mathbb{R}^{13}$, both the PRBP and the values of W stay roughly the same between 1000 and 5000 iterations.

5 | FRAGILITY CURVES

SVM classifiers give to each signal i a score $f_n(X_i)$ whose sign expresses the estimated label, but it does not give directly a probability for this signal to be in one class or the other. In order to define a *fragility curve*, that is, the probability of exceeding the damage threshold as a function of a parameter representing the ground motion, we first need to assign a probability to each signal.

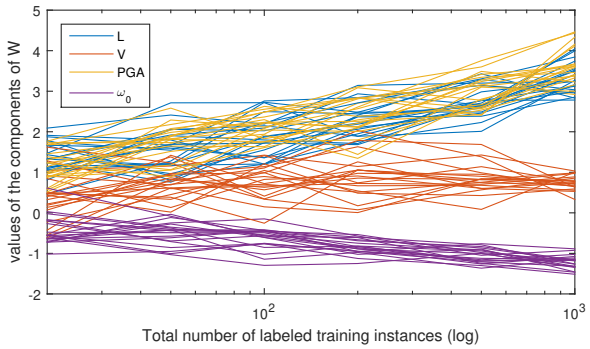


FIGURE 7 Evolution of the 4 main components of W for all 20 test cases.

5.1 | Score-based fragility curve

This probability depends only on the score $f_n(X)$. For a perfect classifier, the probability would be 0 if $f_n(X) < 0$ and 1 if $f_n(X) > 0$; for our SVM classifiers we use a logistic function:

$$p(X) = \frac{1}{1 + e^{-af_n(X)+b}}, \tag{35}$$

where a and b are the slope and intercept parameters of the logistic function (b should be close to 0 if the classifier has no bias, giving a probability of $1/2$ to signals with $f_n(X) \approx 0$). These parameters are computed using a logistic regression on the labeled set $\{(X_{j_1}, l_{j_1}), \dots, (X_{j_n}, l_{j_n})\}$. To compare this estimation with the empirical failure probability of signals with a given score, we divide our database $\mathcal{X}$ in K groups $(I_1, ..., I_K)$ depending on their score, with the k-means algorithm; then we define the estimated and reference probabilities of each group:

$$\begin{aligned} p_k^{est} &= \frac{1}{n_k} \sum_{i \in I_k} p(X_i), \\ p_k^{ref} &= \frac{1}{n_k} \#\left\{ i \in I_k | l_i = 1 \right\}, \end{aligned} \qquad \text{with} \quad n_k = \#I_k. \tag{36}$$

We can now compute the discrete L_2 distance between these two probabilities:

$$\Delta_{L_2} = \sqrt{\frac{1}{N} \sum_{k=1}^{K} n_k (p_k^{ref} - p_k^{est})^2}, \tag{37}$$

with $N = \sum_{k=1}^{K} n_k$. Figure 8 shows this distance for different classifiers using 20, 50, 100, 200, 500 and 1000 labeled instances. The three classifiers (linear SVM in $\mathbb{R}^{13}$, linear SVM in $\mathbb{R}^4$, and RBF kernels in $\mathbb{R}^4$) are compared on 20 test cases, using 20 pairs of starting points (the same for all three). The solid lines show the average L_2 errors, and the dashed lines show the minimum and maximum errors among all test cases.

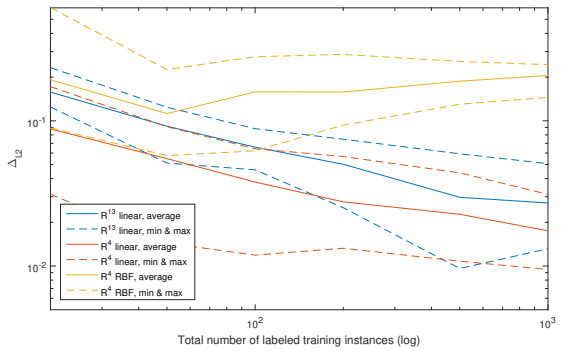


FIGURE 8 Distance between the reference and estimated fragility curves for 3 different active learners.

The average error goes down from 15% after 20 iterations to less than 3% after 1000 iterations for the linear SVM in $\mathbb{R}^{13}$, and from 9% to less than 2% for the linear SVM in $\mathbb{R}^4$. For the SVM using RBF kernels (in yellow in figure 8), the average error does not decrease as the number of iterations increases, and ends up around 20% after 1000 iterations. Figure 9 shows typical examples of fragility curves obtained with each method after 100 and 1000 iterations. Recall that the logistic functions (in red) are not fitted using all the real data (in blue), but only the labeled set, ie 100 or 1000 signals. The linear SVM in $\mathbb{R}^4$ has the least errors in terms of probabilities, although its PRBP is smaller than the linear SVM in $\mathbb{R}^{13}$ when using 1000 labeled instances.

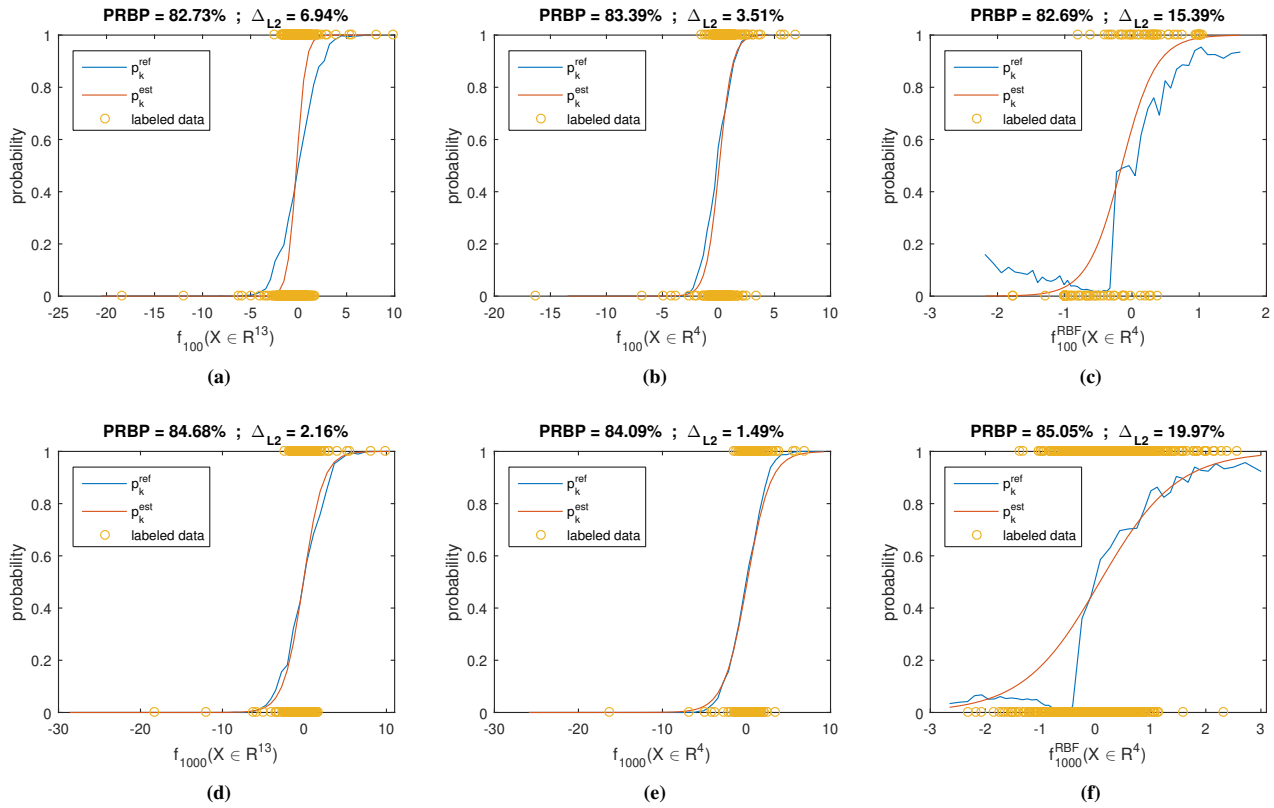


FIGURE 9 Reference and estimated fragility curves using (a and d) a linear SVM in $\mathbb{R}^{13}$, (b and e) a linear SVM in $\mathbb{R}^4$ and (c and f) a RBF SVM in $\mathbb{R}^4$, with (a, b and c) $n = 100$ or (d, e and f) $n = 1000$ labeled instances.

The radial basis function kernel shows a very "strange" behaviour. The probability of failure is not even an increasing function of the score (figures 9 c and 9 f); in particular, signals with a very negative score still have a $5 - 10\%$ chance of exceeding the threshold. This strange shape of p_k^{ref} explains why the Δ_{L_2} error of RBF kernels is so high (figure 8), since we tried to fit a logistic curve on a non-monotonous function. The reason for this major difference between linear and RBF kernels can be understood if we look at the maximum total displacement Z as a function of the score $f_n(X)$, using both kernels (see figure 10). Let us keep in mind that the RBF classifier at 1000 iterations is the most precise of all our active learners; it has the fewest false positives and false negatives of all (see table 1). The sign of the RBF score is thus an excellent predictor for binary classification.

linear kernel	$f(X) < 0$	$f(X) > 0$
$Z > 2Y$	1020	4711
$Z < 2Y$	27175	812

RBF kernel	$f(X) < 0$	$f(X) > 0$
$Z > 2Y$	1009	4722
$Z < 2Y$	27287	700

TABLE 1 Confusion matrix after 1000 iterations for the linear SVM in $\mathbb{R}^4$ (left) and the RBF SVM in $\mathbb{R}^4$ (right).

Figure 10 shows that for the linear classifier, the score is a good predictor of the maximum total displacement Z, with a monotonous relation between the two; therefore the probability that a $Z > 2Y$ is well-approximated by a logistic function of the score. The RBF score, on the other hand, is a poor predictor of the probability of failure, since the relation between the score and the maximum total displacement Z is not monotonous. We can now understand the very high Δ_{L_2} errors for RBF kernels. Looking at figure 10 , we can see that the weakest signals ($Z = 0.005$, just above the elasticity limit) have a RBF score between -1 and -0.4. Since these weak signals are very common in our database, the reference probability p_k^{ref} goes rapidly from 0.5 for $f_n^{\mathrm{RBF}}(X) = 0$ to almost 0 for $f_n^{\mathrm{RBF}}(X) = -0.5$ (see figures 9 c and 9 f), not because the number of positive signals changes significantly between $f_n^{\mathrm{RBF}}(X) = -0.5$ and $f_n^{\mathrm{RBF}}(X) = 0$, but because the number of negative signals is more than 20 times bigger. The linear kernels do not have this problem, and therefore have much lower Δ_{L_2} errors.

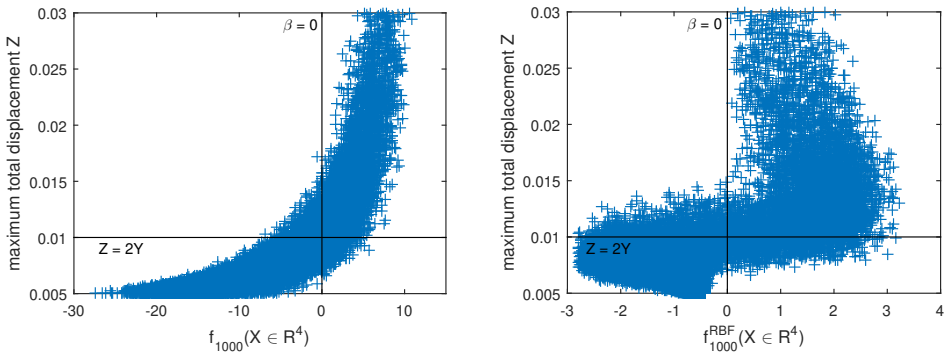


FIGURE 10 Z as a function of the score given after 1000 iterations by (a) the linear SVM in $\mathbb{R}^4$ and (b) the RBF SVM in $\mathbb{R}^4$.

ROC curves (figure 11) give us another way to look at this dilemma between linear and RBF kernels. If we look at the unbiased (i.e. $\beta = 0$) classifiers, the RBF is clearly superior: it has fewer false positives and slightly fewer false negatives than the linear classifier. However, when we choose a negative limit β (see equation 31), for example $\beta = -0.5$, then some of the weakest signals end up over the limit ($f_{1000}^{\mathrm{RBF}}(X) > \beta$) and thus have an estimated label of $\hat{l}(\beta) = 1$. Since these weak signals are so common, the false positive rate becomes extremely high.

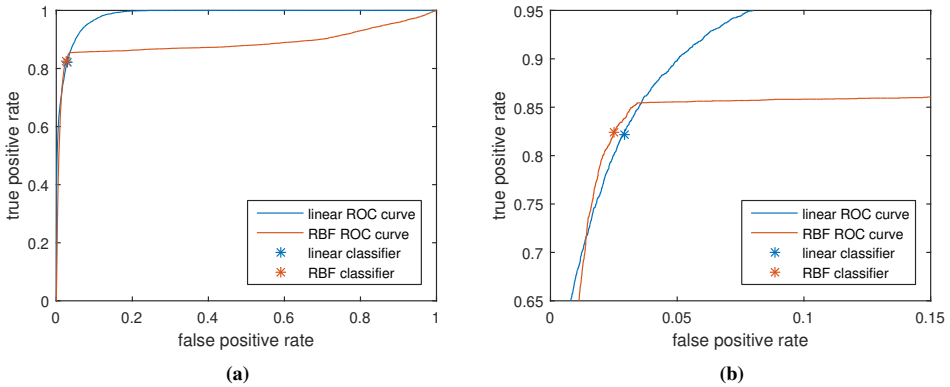


FIGURE 11 (a) ROC curves for two SVM classifiers using linear and RBF kernels, with specific values for the unbiased (i.e. $\beta = 0$) classifiers. (b) zoom on the upper-left corner.

5.2 | PGA-based (resp. L-based) fragility curve

In the previous section we always used the score $f_n(X)$ as the parameter on the x-axis to build the fragility curves. However, our method assigns a probability $p(X)$ to each signal, depending only on a few parameters. If we consider this probability as a function of 4 parameters ($p(L, V, PGA, \omega_0)$ if $X \in \mathbb{R}^4$), then we can use any of those parameters, the PGA for example, to define *a posteriori* a fragility curve depending on just this parameter, averaging over the other ones:

$$p(PGA) = \mathbb{E}[p(X)|PGA]. \tag{38}$$

Figures 12 and 13 show two examples of such curves, using the PGA or the maximum linear displacement L. We used a linear SVM classifier in $\mathbb{R}^4$ with 100 and 1000 iterations and computed the probabilities $p(X)$ as before, but then divided the database in groups (with k-means) depending on their PGA (resp. on L), instead of the score, before computing the reference probabilities p_k^{ref} and estimated probabilities p_k^{est} for each group k. In this case, we can show all 20 test cases in a single figure, since they share a common x-axis (which is not true when we used the score).

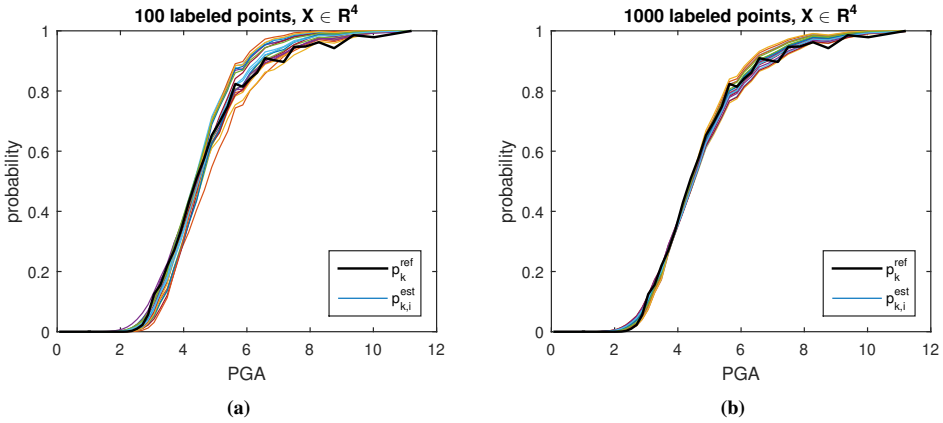


FIGURE 12 Reference and estimated fragility curves as a function of the PGA, using (a) 100 and (b) 1000 labeled points.

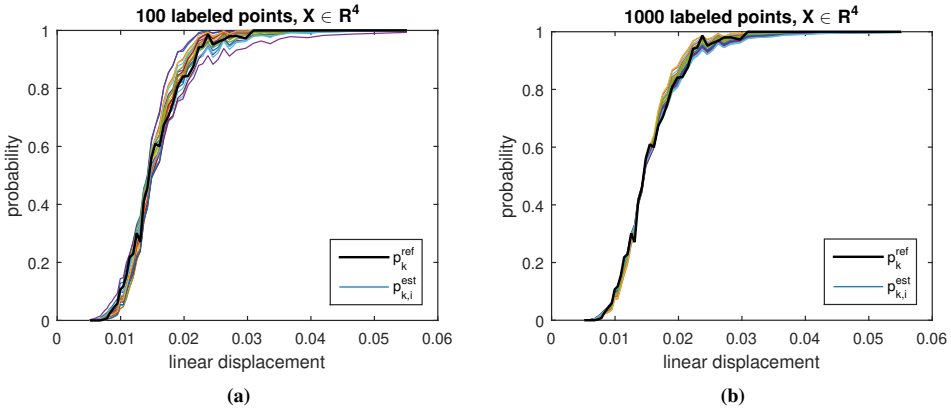


FIGURE 13 Reference and estimated fragility curves as a function of L, using (a) 100 and (b) 1000 labeled points.

We now have a fully non-parametric fragility curve. The distance between the reference and estimated curve is very small in both cases, even using just 100 labeled instances, although the spread is smaller when we add more data points.

5.3 | Trading precision for steepness

The PGA-based and L-based fragility curves (figures 12 and 13) are very close to the reference curves; the distance Δ_{L_2} between reference and estimated curves is very small, even smaller than in the case of score-based fragility curves. In this case, why even bother with score-based fragility curves ? What is their benefit, compared to easily-understandable, commonly accepted PGA-based curves ?

The difference is in the steepness of the curve. Formally, when we construct a fragility curve, we choose a projection $F : \mathbb{R}^4 \mapsto \mathbb{R}$ to use as the x-axis. This projection $F(X)$ can be one of the 4 variables (for example the PGA), or the score $f_n(X)$, which can be a linear or nonlinear (in the case of RBF kernel) combinaison of the 4 variables. We then use the k-means algorithm to make groups of signals who are "close" according to this projection, i.e. signals with the same PGA or the same score; then we compute the estimated probability p_k^{est} for each group. Let us assume for a while that our estimation is very precise, so that $p_k^{est} = p_k^{ref}\ \forall k$. In this case, which fragility curve gives us the most information ? To see this, we define:

$$R^{(F)} = \frac{1}{N}\sum_{k=1}^{K} n_k \phi(p_k^{est(F)}). \tag{39}$$

for some nonnegative-valued function ϕ. Intuitively, a perfect classifier would give each signal a probability of 0 or 1, while a classifier which assigns a probability of $1/2$ to many signals is not very useful. Therefore, we want ϕ to be positive on $(0, 1)$, equal to 0 for $p = 0$ and $p = 1$. If we choose:

$$\phi(p) = -p\ln(p), \tag{40}$$

then $R^{(F)}$ can be seen as the entropy of the probability $p(X)$, which would be equal to 0 for a perfect classifier and has higher values for a "useless" classifier. Another choice would be:

$$\phi(p) = \mathbb{1}_{p\in[0.1,0.9]}. \tag{41}$$

In this case, $R^{(F)}$ also has a clear physical meaning: it is the proportion of "uncertain" signals, i.e. signals such that $p_k^{est}(X) \in [0.1, 0.9]$. Table 2 shows the value of $R^{(F)}$, using the entropy version, for different choices of projection (score, PGA, or linear displacement). We can see on this table that the PGA- and L-based fragility curves are extremely precise, with very low values of Δ_{L_2} (this can also be seen in figures 12 and 13), but their entropy is much higher than the score-based fragility curves.

	projection	score	PGA	L
n=100	Δ_{L_2} (%)	3.8 ± 1.6	2.6 ± 1	2.8 ± 0.9
	entropy (10^{-2})	5.3 ± 1.7	12.3 ± 1.8	12.2 ± 2
n=1000	Δ_{L_2} (%)	1.7 ± 0.6	1.6 ± 0.3	1.4 ± 0.4
	entropy (10^{-2})	7.2 ± 1.2	13.3 ± 1.5	13.6 ± 1

TABLE 2 Precision and entropy of fragility curves using different projections (average and standard deviation over 20 test cases), for n=100 (top) or n=1000 (bottom) labeled points.

One surprising fact of table 2 is that the entropy is smaller at $n = 100$ compared to $n = 1000$ in all three cases. This shows that after only $n = 100$ mechanical calculations, all our classifiers tend to slightly overestimate the steepness, and give fragility curves that are actually steeper than the reality (and also steeper than the more realistic $n = 1000$ curves). This was also seen in figures 9 b and 9 e: at $n = 100$ iterations the estimated curve is steeper than the reference curve, which gives an estimated entropy smaller than the reference entropy. Using the other choice of ϕ gives the same conclusions: the proportion of signals with $p_k^{est} \in [0.1, 0.9]$ is 18.2% if we use the score, but it is around 28% for both the PGA and maximum linear displacement. Therefore, the choice of the projection used for a fragility curve is a trade between precision and steepness. Keep in mind that the values of the entropy for different choices of projection can be obtained after the active learning, and the computationnal cost is very small (mostly the cost of k-means). As a consequence, this choice can be made a posteriori, from the probabilities assigned to each signal.

5.4 | Additionnal remarks

5.4.1 | About the specificity of active learning

Using the score to compute the probabilities (equation 35) on the whole dataset $\mathcal{X}$ is absolutely mandatory, even if one is only interested in the PGA fragility curves. In particular, looking only at the labeled set $\mathcal{L}$ to find directly a probability of failure depending on the PGA gives extremely wrong results. Figure 14 shows not only the reference and estimated fragility curves previously defined, but also the $n = 1000$ points of the labeled set $\mathcal{L}$ and an empirical probability built from it.

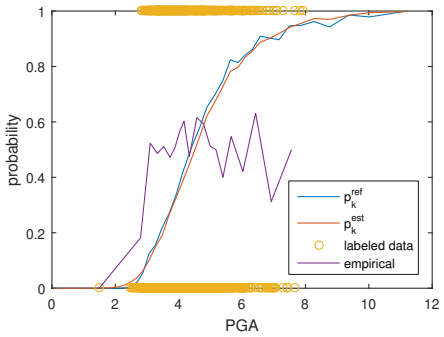


FIGURE 14 Empirical fragility curve using only the labeled set (violet).

This empirical fragility curve was computed by using the k-means algorithm on the PGA values of the labeled set $\mathcal{L}$, then taking the empirical failure probability in each group. The result looks like... a constant around 0.5 (with high variability). Looking only at $\mathcal{L}$, the PGA does not even look correlated with failure. The reason for this surprising (and completely wrong) result is the active learning algorithm. $\mathcal{L}$ is not at all a random subset of $\mathcal{X}$; it is a carefully chosen set of the data points with maximal uncertainty, which means that most of them end up very close to the final hyperplane (and have a final score very close to 0), but also that they span as much of this hyperplane as possible. When these points are projected on any axis not perpendicular to the hyperplane (that is, any axis but the score), for example the PGA (which is one of the components of X), the empirical probability is roughly equal to the constant $1/2$, which is not representative at all of the underlying probability. Using equation 35 (then eventually fitting the results to a lognormal curve for the PGA) is the right way to go.

5.4.2 | About a combination of the linear and RBF kernels

The RBF kernel was very promising in terms of classification, as we saw on the first results (figure 5 a); however, we saw in the following sections that using it to make a fragility curve can lead to catastrophic results (figures 9 c and 9 f). Could we combine the two kernels in a way that let us keep the benefits of both ? One simple way to do that is to use the following procedure:

- use the active learning with RBF kernel to select $n = 1000$ signals to be labeled;
- from these 1000 data points, train two classifiers, one with a linear kernel, the other with the RBF kernel;
- assign two scores $f_{lin}(X)$ and $f_{rbf}(X)$ to each non-labeled signal, using the two classifiers;
- fit the labeled points to each set of scores, giving you two probabilities $p_{lin}(X)$ and $p_{rbf}(X)$ for every signal;
- the "final" probability is chosen as:

$$p(X) = \begin{cases} p_{lin}(X) \text{ if } \quad p_{lin}(X) < 0.05 \text{ or } p_{lin}(X) > 0.95, \\ p_{rbf}(X) \text{ otherwise.} \end{cases} \tag{42}$$

Since the active learning used RBF kernels and we use the RBF score for any "uncertain" signal, the PRBP has about the same value than in the "pure RBF" version, around 0.85. However, the Δ_{L_2} is at 2.3%, slightly higher than for the "pure linear" version (1.8%), but much better than the catastrophic "pure RBF" version (20% !). Although this procedure may seem to have

the best of both worlds, in a practical application the PRBP may not be very interesting if the goal is to make a fragility curve; in this case only the precision and steepness of the curve are important, and a linear kernel performs better than a RBF kernel.

6 | CONCLUSION

This paper proposed an efficient methodology for estimating non-parametric seismic fragility curves by active learning with a Support Vector Machines classifier. We have introduced and studied this methodology when aleatory uncertainties have a predominant contribution in the variability of structural response, that is to say when the contribution of uncertainties regarding seismic excitation is "much larger" than the contribution of uncertainties regarding structural capacity. In this work, structure was considered as deterministic. In this framework, a perfect classifier, if it exists, would lead to a fragility curve in the form of a unit step function, i.e. corresponding to a fragility curve "without uncertainty". That means the output of this classifier, which is a score, would be the best seismic intensity measure indicator to evaluate the damaging potential of the seismic signals, knowing that such a classifier would necessary be both structure and failure criterion-dependent, with possibly a dependence on the ground motion characteristics (near-fault type like, broadband, etc).

The proposed methodology makes it possible to build such a (non-perfect) classifier. It consists in (i) reducing input excitation to some relevant parameters and, given these parameters, (ii) using a SVM for a binary classification of the structural responses relative to a limit threshold of exceedance. Selection of the mechanical numerical calculations by active learning dramatically reduces the computational cost of construction of the classifier. The output of the classifier, the score, is the desired intensity measure indicator which is then interpreted in a probabilistic way to estimate fragility curves as score functions or as functions of classical seismic intensity measures.

This work shows that a simple but crucial preprocessing of the data (i.e. Box-Cox transformation of the input parameters) makes it possible to use a simple linear SVM to obtain a very precise classifier after just one hundred iterations, that is to say with one hundred mechanical calculations. Moreover, for the class of structures considered, with only four classical seismic parameters (PGA, V, L, ω_0), the score-based fragility curve is very close to the reference curve (obtained with a massive Monte Carlo-based approach) and steeper than the PGA-based one, as expected. L-based fragility curves appear to perform about as well as PGA-based ones in our setting. Advanced SVMs using RBF kernel result in less classification errors when using one thousand mechanical calculations, but does not appear well suited to making fragility curves.

A naive way to take into account epistemic uncertainties would consist in building a classifier for each set of structural parameters. Nevertheless, such a method would not be numerically efficient. Another way could be to assume that epistemic uncertainties have small influence on the classifier evaluated, for example, for the median capacity of the structure. Thus, only calculations of linear displacements would be necessary to estimate the corresponding fragility curve. However, to avoid such assumptions, some research efforts have to be devoted to propose an efficient overall methodology that takes into account the two types of uncertainties.

References

1. Zhang Jian, Huo Yili. Evaluating effectiveness and optimum design of isolation devices for highway bridges using the fragility function method. *Engineering Structures.* 2009;31(8):1648–1660.

2. Saha Sandip Kumar, Matsagar Vasant, Chakraborty Subrata. Uncertainty quantification and seismic fragility of base-isolated liquid storage tanks using response surface models. *Probabilistic Engineering Mechanics.* 2016;43:20–35.

3. Patil Atul, Jung Sungmoon, Kown Oh-Sung. Structural performance of a parked wind turbine tower subjected to strong ground motions. *Engineering Structures.* 2016;120:92–102.

4. Gidaris Ioannis, Taflanidis Alexandros A, Mavroeidis George P. Kriging metamodeling in seismic risk assessment based on stochastic ground motion models. *Earthquake Engineering & Structural Dynamics.* 2015;44(14):2377–2399.

5. Wang F, Feau C. Seismic fragility curve estimation using signals generated with GMPE - case study on the Kashiwazaki-Kariwa power plant. In: SMiRT-24; 2017; Busan, Korea.

6. Quilligan A, O'Connor A, Pakrashi V. Fragility analysis of steel and concrete wind turbine towers. *Engineering Structures.* 2012;36:270–282.

7. Der Kiureghian Armen, Ditlevsen Ove. Aleatory or epistemic? Does it matter?. *Structural Safety.* 2009;31(2):105–112.

8. Kennedy RP, Cornell CA, Campbell RD, Kaplan S, Perla HF. Probabilistic seismic safety study of an existing nuclear power plant. *Nuclear Engineering and Design.* 1980;59(2):315–338.

9. Zentner I. Numerical computation of fragility curves for NPP equipment. *Nuclear Engineering and Design.* 2010;240(6):1614–1621.

10. JW Baker. Efficient analytical fragility function fitting using dynamic structural analysis. *Earthquake Spectra.* 2015;31:579–599.

11. Mai Chu, Konakli Katerina, Sudret Bruno. Seismic fragility curves for structures using non-parametric representations. *Frontiers of Structural and Civil Engineering.* 2017;11(2):169–186.

12. Karamlou Aman, Bocchini Paolo. Computation of bridge seismic fragility by large-scale simulation for probabilistic resilience analysis. *Earthquake Engineering & Structural Dynamics.* 2015;44(12):1959–1978.

13. Park Joonam, Towashiraporn Peeranan. Rapid seismic damage assessment of railway bridges using the response-surface statistical model. *Structural Safety.* 2014;47:1–12.

14. Seo Junwon, Linzell Daniel G. Use of response surface metamodels to generate system level fragilities for existing curved steel bridges. *Engineering Structures.* 2013;52:642–653.

15. Wang Zhiyi, Pedroni Nicola, Zentner Irmela, Zio Enrico. Seismic fragility analysis with artificial neural networks: Application to nuclear power plant equipment. *Engineering Structures.* 2018;162:213–225.

16. Hasenjäger M, Ritter H. Active Learning in Neural Networks:137–169. Heidelberg: Physica-Verlag HD 2002.

17. Seung HS, Opper M, Sompolinsky H. Query by Committee. In: COLT '92:287–294ACM; 1992; New York, NY, USA.

18. Gazut S, Martinez JM, Dreyfus G, Oussar Y. Towards the Optimal Design of Numerical Experiments. *Trans. Neur. Netw.*. 2008;19(5):874–882.

19. Tong Simon, Koller Daphne. Support Vector Machine Active Learning with Applications to Text Classification. *J. Mach. Learn. Res.*. 2002;2:45–66.

20. Rezaeian Sanaz, Der Kiureghian Armen. Simulation of synthetic ground motions for specified earthquake and site characteristics. *Earthquake Engineering & Structural Dynamics.* 2010;39(10):1155–1180.

21. Sanaz Rezaeian. Stochastic modeling and simulation of ground motions for performance-based earthquake engineering. PhD thesisUniversity of California, Berkeley2010.

22. Ambraseys NN, Smit P, Berardi R, Rinaldis D, Cotton F, Berge C. *Dissemination of European StrongMotion Data.* CD-ROM collection. European Commission, Directorate-General XII, Environmental and Climate Programme, ENV4-CT97-0397, Brussels, Belgium; 2000.

23. Adler Robert J. *The Geometry of Random Fields*. Philadelphia: SIAM; 2010.

24. Kristan Matej, Leonardis Aleš, Skočaj Danijel. Multivariate online kernel density estimation with Gaussian kernels. *Pattern Recognition.* 2011;44(10):2630–2642.

25. Kremer Jan, Steenstrup Pedersen Kim, Igel Christian. Active Learning with Support Vector Machines. *Wiley Int. Rev. Data Min. and Knowl. Disc.*. 2014;4(4):313–326.

How to cite this article: Sainct R., Feau C., Martinez J-M., and Garnier J. (2018), Efficient Seismic fragility curve estimation by Active Learning on Support Vector Machines, *journal*, *volume*.