Critical analysis of the suitability of surrogate models for finite element method application in catalog-based suspension bushing design.

CERNUDA, C., LLAVORI, I., ZĂVOIANU, A.-C., AGUIRRE, A., ZABALA, A. and PLAZA, J.

2020

Critical Analysis of the Suitability of Surrogate Models for Finite Element Method Application in Catalog-Based Suspension Bushing Design

Carlos Cernuda
Electronics and Computing Dept.
Mondragon University
Arrasate-Mondragon, Spain
ccernuda@mondragon.edu

Iñigo Llavori
Surface Technologies
Mondragon University
Arrasate-Mondragon, Spain
illavori@mondragon.edu

Alexandru-Ciprian Zăvoianu
School of Computing Science and Digital Media
Robert Gordon University
Aberdeen, UK
c.zavoianu@rgu.ac.uk

Aitor Aguirre
Electronics and Computing Dept.
Mondragon University
Arrasate-Mondragon, Spain
aaguirre@mondragon.edu

Alaitz Zabala
Surface Technologies
Mondragon University
Arrasate-Mondragon, Spain
azabalae@mondragon.edu

Jon Plaza
R&D Antivibration/Sealing Department
Cikatek R&D and Innovation Centre
Berriatua, Spain
jplaza@cikatek.com

***Abstract*—This work presents a critical analysis of the suitability of surrogate models for finite element method application. A case study of a finite element method (FEM) structural problem was selected in order to test the performance of surrogate algorithms. A simple design of experiments (DoE) approach, based on 1D kernel density estimations, is employed to construct a representative pool of real FEM simulations, which becomes the dataset for five different surrogate models, two linear and three non-linear, whose most relevant hyperparameters were tuned (model selection). Results in a real bushing case study show that surrogate models can accurately mimic FEM simulations outcomes, in this case four types of stiffnesses (axial, radial, torsion, and cardanic).**

***Index Terms*—Design of experiments, Surrogate model, Finite element method, Bushing, Support vector regression, Random Forest**

I. Introduction

The finite element method (FEM) is the most popular numerical method to solve partial differential equations. The method is based on simplifying a large, complex and continuous problem into an approximate composed of simple, discrete small parts called finite elements [1]. The architecture of the FEM formulation is based in a system of algebraic equations where each finite element is assembled into a large system that models the behaviour of the original problem. FEM applications are widespread, ranging from structural problems [2], to more advance ones such as the fluid-structure iteration (FSI) [3] [4].

However, one major drawback of the FEM is that the execution time needed to solve the partial differential equations is large. In an industrial application framework, usually lots of different try-outs are needed to obtain a result that meets all requirements, i.e. variations of the original problem are introduced in order to evaluate the impact of existing variables. Therefore, the lead time to obtain a desired result increases considerably and this directly translates into an increased final product cost. Furthermore, the optimal result usually is never achieved as the time needed to obtain it is frequently prohibitive.

Machine learning could be defined as *the field of study that gives computers the ability to learn without being explicitly programmed* (Arthur Samuel, 1959). Every machine learning model, in its own particular way, will learn underlying relations inside the data it has access to, without being guided by any external influence, apart from the definition of the hyperparameters that might define the model itself. In fact, many machine learning algorithms (such as Takagi-Sugeno fuzzy systems [5] or artificial neural networks among others) are universal function approximators [6] [7] [8], meaning that they would eventually asymptotically mimic the output of any existing function. Therefore, if we understand the FEM as a function, whose output is the approximate solution of the partial differential equations, we can be sure that with enough data and a possibly asymptotically infinite complex model we would be able to exactly replicate the FEM. The key question is if we are able to relax the needed data and the complexity level in such a way that the loss in accuracy of our model is low enough to get high quality approximations of the FEM's output in a reasonable time frame (*surrogate model*).

In this work, a suspension bushing component is employed as a case study of a general FEM problem to analyze the suitability of surrogate models. Bushings are usually employed in vehicles suspension systems as anti-vibration components. They are made of rubber and are enclosed with two concentric metallic parts that are press-fitted from the inner and outer

Department of Economic Development and Infrastructures of the Government of the Basque Country, under the agreement KK-2019/00022.

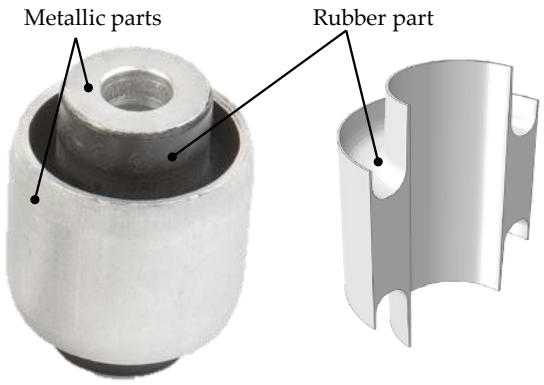

Fig. 1. Example of a bushing component where a cross section of the rubber part is shown. Adapted figure from [12].

rings (see Fig. 1). The overall structural behavior of bushings is difficult to predict, as the rubber is a hyper-elastic material that undergoes large strains. Due to the aforementioned problem, a large number of different trials are needed to fine-tune the design of each bushing and this has associated impact in the cost of the product. Bushing design using FEM has been considered in the literature [9] [10] [11]. In this context, a surrogate model might bring a fast alternative to the typical FEM-based procedure, accelerating considerably the lead-time in the design phase.

Therefore, this work presents a critical analysis of the suitability of surrogate models for FEM application, using the bushing component as case study. For that, the rest of the paper is organized as follows. In Section II the finite element analysis is theoretically described, and the specific bushing case study used in the experiments put into context. Next, Section III defines surrogate models, as well as our approach combining the design of experiments (DoE) with machine learning surrogate models for FEM simulations. Then, Section IV shows the application of DoE to our problem (IV-A), presents the considered types of machine learning algorithms for regression problems (IV-B), and fully describes the whole experimentation performed, including the variants of the algorithms used, the hyperparameters to be tuned for each of them, and the exact way to train and evaluate the developed surrogate models (IV-C). Furthermore, Section V presents the results achieved and a complete discussion about them. Last, Section VI states the conclusions and propose possible promising future lines.

II. Finite Element procedure

Finite element analysis is frequently divided into three blocks. The first one is the so called pre-processing where the geometry, meshing and the boundary conditions among others are defined. Usually this process is done manually by the engineer in a commercial finite element software suit such as Abaqus FEA. The second block is the processing, where the partial differential equations are solved. Finally, the last block is the post-processing where the results are analysed by the engineer.

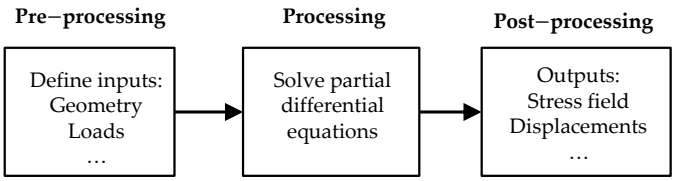

Fig. 2. Finite element method procedure.

In order to develop a surrogate model, an automatic pre- and post- process needs to be developed in order to perform hundreds or thousands of numerical simulations. In this work, a parametric scripting was developed in Python where the whole finite element process was automatised (see Fig. 2). This way, numerous different geometries and loading conditions can be tested, in order to generate the needed data to construct the surrogate models.

Subplots (a)-(d) from Fig. 3 illustrate the four different FEM simulations we performed in accordance with the description in [12]: from theses simulations, the stiffness of each case (k) can be easily calculated from the applied force (F) or moment (M) vs. displacement (δ) or rotation (θ) plot (see Fig. 3e):

$$k = F/\delta, \tag{1}$$

$$k = M/\theta. \tag{2}$$

Therefore, the above mentioned parametric FEM procedure can be used to generate the raw data needed to develop a surrogate model, i.e. relating the inputs (geometry, material constitutive model, applied forces among others) with the outputs (stiffness values).

III. Efficient design of surrogate models

Formally, a *surrogate model* could be defined as a meta-model that is capable of mimicking, as closely as possible, the behavior of the original model that is being surrogated while being computationally cheaper to evaluate. Computational cost could be understood in several ways, e.g. as computational resources or time. In engineering, surrogate models are used for approximating an outcome of interest in case this is time-consuming, expensive, or somehow difficult to be measured.

As mentioned before, in the presence of enough data and having theoretically unlimited memory, computational power and time, any function, including FEM simulations, could be approximated by certain machine learning models. Our aim is to (i) first perform a Design of Experiments, in order to get a small but diverse pool of simulations, complete enough for surrogate modelling to be feasible on them. Then, (ii) tune the hyperparameters of promising regression modeling techniques to find low complexity final surrogate models that are precise enough that their outcomes can be safely used as replacements for FEM simulations.

The objective of the DoE is to end up with a real FEM simulations dataset, as small as possible, containing enough informative power. In order to construct our dataset, we

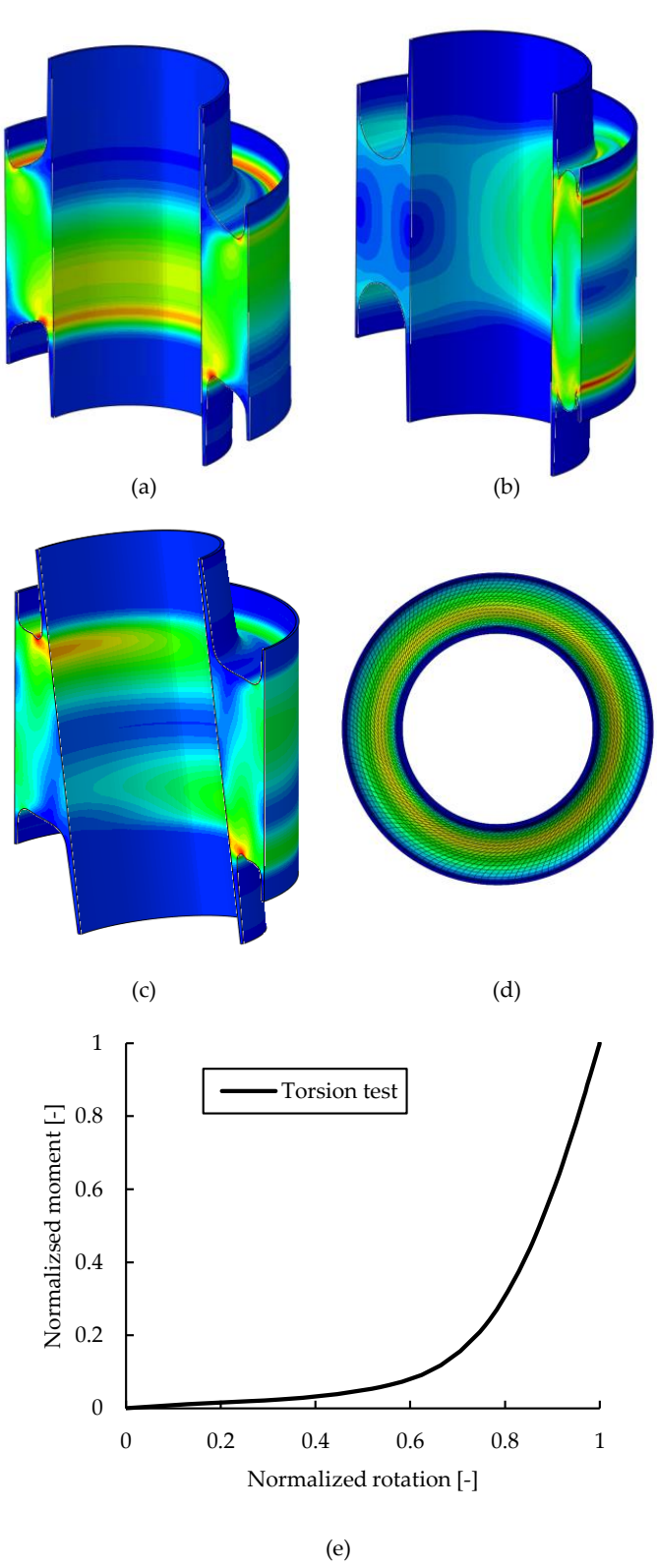

Fig. 3. von Mises stress field for each static simulation– (a) axial, (b) radial, (c) cardanic, (d) torsion (upper view), and (e) applied force vs. displacement plot (note that the axis values were normalised due to industrial confidentiality purpose.

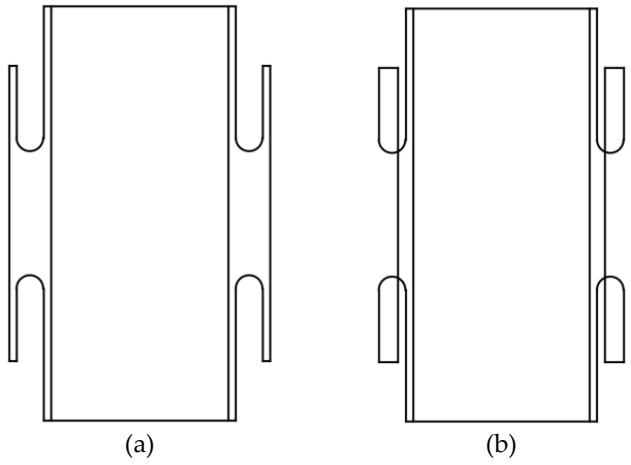

Fig. 4. Example of feasible (left) and unfeasible (right) bushing configurations.

assume that we have a catalog with several parametrizations (inputs for the FEM) of interest. In this way, we can know the actual geometric and loading requirements and also have a general view of the type of FEM simulations that are being performed. We must take into account that the FEM simulations could stop without completing all forces ranges for numerical reasons, mainly in cases of high non-linearity. In such cases, despite the surrogate model would provide a prediction, that prediction must be considered as uncertain. Moreover, the catalog is assumed to contain information about the *catalog control variables*, i.e. features whose values must be fixed because they are static design limitations (e.g. the outer and inner rings radii in Fig. 1), so that we have access to the current design limitations with a double purpose: (i) polishing current designs, and (ii) being prepared for potential new requirements. Consequently, ranges for each feature in the parametrization are implicitly provided. Then, we would try to cover, with certain density, the hypercube in the feature space given by them. As the hypercube is a Cartesian product of feature ranges, where each feature contributes independently, we also proceed in a feature-wise way (i.e. without *control variables*). Nevertheless, there might be certain feature combinations that are physically unfeasible, enforcing the definition of design constraints. Fig. 4 shows examples of one feasible (left) and one unfeasible (right) configuration.

Instead of filling the feasible regions of the hypercube uniformly, we perform a 1D kernel density estimation, using a Gaussian kernel, for each feature in the catalog. Then, we adapt the density of the coverage of the dataset to be constructed according to the estimated density. Notice that the Gaussian kernel guaranties a minimum coverage in the whole range, as it is always positive. Nevertheless, as sometimes that minimum coverage could be too small, a threshold for the minimum coverage could be imposed. Fig. 5 and Fig. 6 show two real example of the KDE for two of the variables of torsion data, one discrete geometric feature, and one continuous force

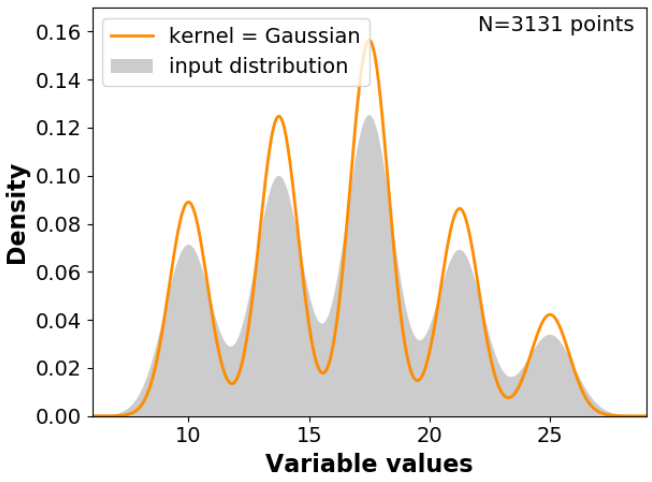

Fig. 5. Real example of 1D kernel density estimation with a Gaussian kernel. Due to the limited amount of unique values, the real density could be calculated, and is shown in grey. It corresponds to a geometric discrete variable of torsion data, whose values are not depicted. Nevertheless, the true density allows us to deduce that their positions correspond to 5 concrete zones of influence identified by the peaks. The values in the horizontal axis have been modified for confidentiality issues.

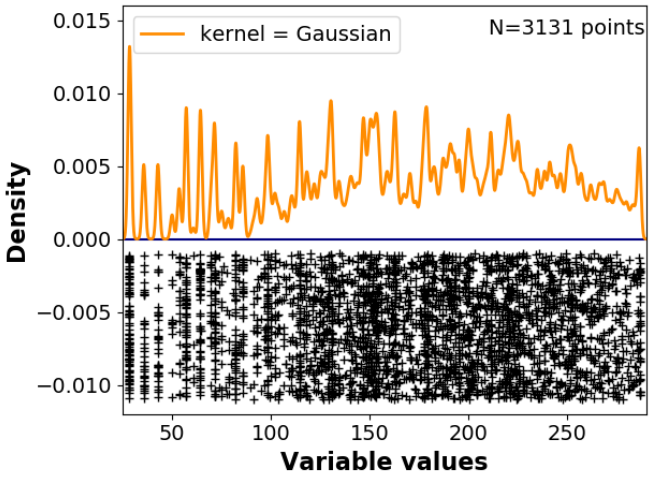

Fig. 6. Real example of 1D kernel density estimation with a Gaussian kernel. It corresponds to a continuous force variable of torsion data. The black crosses are the 3131 values (placed horizontally), distributed vertically for visualization purposes. The values in the horizontal axis have been modified for confidentiality issues.

feature.

IV. Experimental setup

A. Design of experiments

The reference catalog we will employ has six different designs, defined by six pairs of the two control variables, *external radius* and *internal radius*, corresponding to the outer and inner rings' radii respectively. For each of them, we have several feature-level design constrains, that limit the possible values of both the geometric and force variables. The real examples in Fig. 5 and Fig. 6 illustrate graphically the type of available information. Using general product-level design constraints, and extracting the extreme values and densities of all variables, we have generated an adequate grid of possible configurations to be simulated for constructing our dataset, keeping a coverage of the whole configuration space that reflects the actual coverage of the configurations catalog, and respects the minimum general coverage threshold.

The complete dataset can not be shown for confidentiality issues. Table I shows a summary of the generated datasets, one for each of the four stiffness targets of interest. Our aim is to be able to predict the FEM output stiffness measures, not only when considering minor alterations to the already known six product types, but also for hypothetical new types whose control variables are reasonably similar to the known ones.

TABLE I
Datasets description

Stiffness	Designs	CtrlVars	GeomVars	Forces	Samples
Axial	6	2	6	1	1080
Radial	6	2	6	1	1340
Torsion	6	2	6	1	3131
Cardanic	6	2	6	1	1799

B. Regression algorithms

Among all possible regression algorithms we could have considered, the ones we have chosen are support vector regression (linear and non-linear variants), random forest (non-linear), and generalized linear models with elastic net (linear with regularization). These algorithms have, at the same time, relatively low complexity (compared to highly complex algorithms such as deep learning approaches) and potentially high predictive capabilities (flexibility).

a) Support Vector Regression: Support vector machines (SVM) [13] is a well-known classification method, based on separating the classes employing hyper-planes is such way that the separation is maximized (optimization problem). This separation is not performed in the original input space but in a kernel-transformed space, i.e. the kernel trick [14]. The samples that are closest to the decision boundary, thus defining the hyper-planes, are called *support vectors*. Depending on the type of kernel function employed, the resulting model could be linear or non-linear. Depending on both the kernel and how the underlying optimization problem is defined, there are several SVM versions. The same philosophy is kept for regression, adapting the idea of maximum separation to numerical instead of categorical output data. The usual kernels are linear, and Gaussian radial basis function (RBF) defined by means of a parameter called γ. The regular SVM algorithm employs a *cost* parameter, meant for controlling the errors in the definition of the hyper-planes, and ν-SVM [15] substitutes the cost by a parameter ν that is able to limit the amount of support vectors and thus the complexity of the model. The adaptation of support vector machines for regression problems is called support vector regression (SVR) [16].

b) Generalized Linear Models with Elastic Net: Generalized linear models [17] are a generalization of ordinary linear regression that provides flexibility in the sense that the distribution of the errors is not necessarily assumed to be normal, as with ordinary linear regression. The combination of the elastic net with generalized linear models (GLMnet) is a regression algorithm based on generalized least squares that uses cyclical coordinate descent [18] in a path-wise fashion [19] in order to select the optimum elasticity in the regularization via the elastic net. The elasticity is provided by the possibility of controlling, by means of a convex linear combination, how close we are to Lasso ($L1$ penalty) or ridge regression ($L2$ penalty) using a single parameter, called α, so that a 0 value means pure $L2$, a value 1 means pure $L1$, and a value between them leads to a combined penalty. This allows an efficient exploitation of the regularization benefits. The amount of regularization that is performed is also controlled, by means of the regularization parameter λ.

c) Random Forest: Random forest (RF) algorithm [20] is a stochastic ensemble method that performs a bagging strategy by combining bootstrapping (with replacement) and aggregation of decision trees (weak learners). The procedure relies on training a preset number n of trees, with the particularity of using only a subpopulation of the available samples, chosen by bootstrapping, for training. By default, the size of the subpopulations is the same as the one of the original population, thus the expectation is that 68.27% of the samples will be included in each subpopulation [21], being the rest simply replicates. Moreover, at each node of each tree, only a subset of the original features is randomly selected for deciding how to split the current branch into two subbranches. By default, $\sqrt{M}$ are used, where M is the number of features. Given that decision trees tend to overfit as their depth increases, RF attempts to tackle this phenomenon by limiting the depth of each tree through a parameter. When the model is applied to a new sample for validation, the decision is obtained by combining all individual n tree decisions by aggregation (majority voting in classification problems, and averaging in regression problems).

C. Experimental scheme

In the experiments, we have tried to exploit the capabilities of each of the regression algorithms by fine tunning their most relevant hyperparameters. In order to validate the models, the data has been split into train (75%) and test data (25% of the total data samples). As randomness is involved in the split process, we conducted several repetitions to diminish the influence of chance. We can report that the obtained results are very stable as there is low variance when experimenting on different training-test random splits of the total data. The chosen strategy for model selection for each algorithm has been best performance over a *grid search* (GS) combined with *10-folds cross validation* (CV). In grid search all possible combinations of all the selected options for the parameters in the grid are tried. Then, for each combination, CV is performed, consisting on partitioning the data into 10 subsets and train and test 10 models using sequentially one fold for testing and the other 9 for training.

The parameter grids defined for each algorithm are the following:

a) Linear support vector regression (LSVR): The hyperparameters to be tuned and the considered options for them are the cost (C) and the zero-loss margin (ϵ), which is a threshold under which errors are neglected. The grid, comprising 66 combinations, is defined by the options:

$$\begin{cases} C \in \left\{2^{-5}, 2^{-3}, \ldots, 2^{15}\right\} \\ \\ \epsilon \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.15\}\,. \end{cases} \tag{3}$$

Despite LSVR being deterministic by definition, the order of the samples might influence the result because of the way the calculations are performed. Therefore, we have run ten times each GS CV with different sample shuffling trials, averaging the scores. Therefore the total amount of trials is 660.

b) Support vector regression (SVR): The hyperparameters and options considered are the cost (C) and the zero loss margin (ϵ), with the same options as in LSVR, and the γ parameter that defines the RBF kernel. The final parameter search comprises 660 combinations, considering the options for γ:

$$\gamma \in \left\{2^{-15}, 2^{-13}, \ldots, 2^{3}\right\}. \tag{4}$$

c) Nu support vector regression (NuSVR): The hyperparameters and options considered are C and γ, with the same options as in SVR, and the ν that controls the final amount of support vectors. The grid comprises 396 combinations, considering the options for ν:

$$\nu \in \{0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}\,. \tag{5}$$

d) Generalized linear models with elastic net (GLMnet): In this case, we have selected the previously described α and λ parameters, controlling respectively the intensity of the regularization and the penalty type balance. The grid, comprising 610 combinations, is defined by

$$\begin{cases} \alpha \in \{0.01\} \cup \{0.05, 0.1, \ldots, 1.0\} \cup \{1.1, 1.2, \ldots, 5.0\} \\ \\ \lambda \in \{0.1, 0.2, \ldots, 0.9, 1.0\}\,. \end{cases} \tag{6}$$

e) Random forest: For RF, the hyperparameters we have tuned are meant for providing learning capacity through horizontal complexity increase (number of trees, n), and for controlling the risk of overfitting by limiting the vertical complexity (maximum tree depth, $maxDepth$), and by limiting the internal information per node (maximum seen features, $maxFeat$). The grid comprises 64 combinations, given by

$$\begin{cases} n \in \{10, 50, 100, 150, 200, 500, 750, 1000\} \\ \\ maxDepth \in \{\infty, 20, 10, 5\} \\ \\ maxFeat \in \left\{\sqrt{M}, M\right\}, \end{cases} \tag{7}$$

where M is the total amount of features. There are also certain stochastic processes inside the RF algorithm, concretely in the random selection of the features to use in each splitting node of the trees, so we have run ten times each GS CV, averaging the scores. Therefore the total amount of trials is 640. Both here and in LSVR, the results of the repetitions are pretty similar, thus 10 times seems to be a safe amount.

V. Results

As mentioned in Section IV-A (Table I), there are 6 different designs for each stiffness. It makes sense to consider two scenarios where one of the existing designs is improved, or a new design is introduced in the catalog. For the latter, it makes more sense to ignore the control variables in the surrogate model definition, focusing on the modifiable features. In any case, the new control variables should not be in the extrapolation zones of the control variables space if we want our predictive surrogate model to be precise.

Tables II and III show the results, with R^2 as quality score, using all features and all features excluding the control variables respectively. For each algorithm and each stiffness, we have a double line containing both the CV score (average R^2 over the 10 CV test folds), and the test R^2 score on the 25% of reserved (i.e., unseen by the CV) test data, for the best hyperparameters in the corresponding algorithm grid. In the case of the algorithms with repetitions due to their stochastic content (LSVR and RF), the results on the table are the averages of the 10 repetitions. Finally, the last column shows the average scores across all four stiffness targets. The scores that are statistically significantly worse than the best for each stiffness (marked in bold), using Wilcoxon signed-rank test [22], are marked with a star.

TABLE II
Results including the 2 control variables.

		Stiffness				
		Axial	**Radial**	**Torsion**	**Cardan.**	**Aver.**
LSVR	**CV**	.83411*	.6029*	.1631*	.4559*	.5140
	Test	.83703*	.5719*	.1642*	.3949*	.4920
SVR	**CV**	**.99997**	**.9987**	.1322*	.8282	.7398
	Test	**.99995**	**.9997**	.1478	.6793*	.7067*
NuSVR	**CV**	**.99997**	**.9987**	.1348*	.8283	.7404
	Test	.99994	**.9997**	.1439*	.6786*	.7055
GLMnet	**CV**	.84308*	.6982*	.3933*	.5127*	.6077
	Test	.84693*	.6780*	.3649*	.4504*	.5851
RF	**CV**	.99942	.9983	**.9976**	**.9746**	**.9925**
	Test	.99967	.9993	**.9985**	**.9882**	**.9964**

First, for the case including all features, we have to point out the problems of all three SVR variants to converge in torsion stiffness, even diminishing the tolerance parameter to allow the algorithms to run longer towards convergence. The other two algorithms, GLMnet and RF, show very different performances, due to linearity. The same differences between linear and non-linear methods can be observed for the other stiffness targets: non-linear methods generally perform much better than linear ones. This is expected, because of the exponential behavior of the stiffness (see Fig. 3(e)). Both non-linear SVR methods always behave very similarly in light of the similarities between their hyperparameters grids.

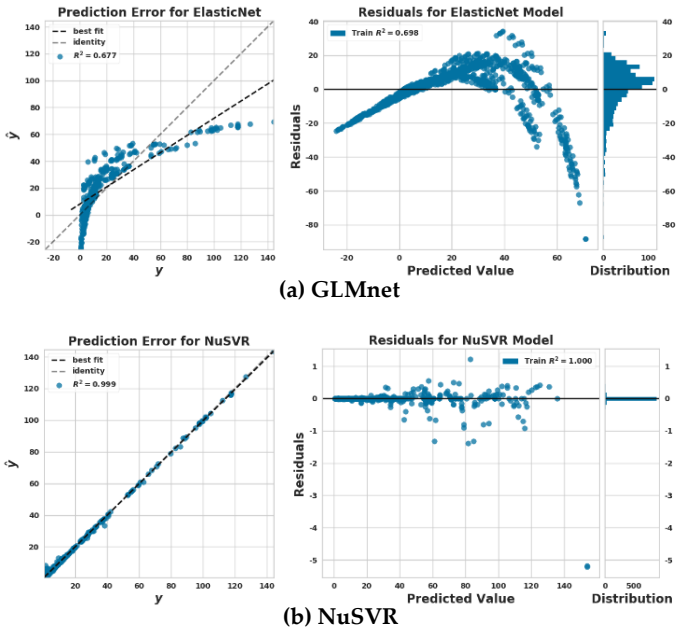

Fig. 7. Prediction error plot (left) and residual plots with distribution histograms (right) for axial stiffness prediction including control variables. Both linear methods on one hand and all three non-linear methods on the other behave similarly. Therefore, we show just the best examples of linear (GLMnet), and non-linear (NuSVR) surrogate models.

Comparing CV and test results across all cases, there is strong empirical evidence that the results are not impacted by overfitting as the test scores are not much worse than the CV (train) ones. In general, RF displays the best performance as: (i) it achieves near-best results on the axial and radial stiffness targets given that the difference of top performing SVR and NuSVR is not statistically significant, (ii) it outperforms by a large margin all other algorithms on the torsion and cardanic stiffness targets (see Tables II and III).

As an illustrative example, Fig. 7 shows both the prediction error plot, and residual plots with distribution histograms, for radial stiffness prediction including control variables. Only the best linear and non-linear methods are depicted. We can clearly see that the linear method, GLMnet in Fig. 7(a), suffers when predicting the extreme values, being much worse in the high values, that are known to be associated with more non-linear effects – see Fig. 3(e). On the contrary, the non-linear method, NuSVR in 7(b), is stable in the whole target range, being able to capture the non-linear essence of the stiffness very well.

In the second case, without using the control variables, the results are generally similar, but with slightly worse R^2 scores. A possible reason, could be that now the algorithms do not have the chance to differentiate between different catalog designs and behave as if they are making predictions based only on samples originating from the same design.

TABLE III
RESULTS EXCLUDING THE 2 CONTROL VARIABLES.

		Stiffness				
		Axial	**Radial**	**Torsion**	**Cardan.**	**Aver.**
LSVR	**CV**	.7375*	.5310*	.0049*	.3216*	.3987
	Test	.7273*	.5162*	.0138*	.2960*	.3883
SVR	**CV**	.9193	.8587	.0461*	.6441*	.6171
	Test	.8798	.9198	.0624*	.6574*	.6299
NuSVR	**CV**	**.9195**	.8587	.0817*	.6478*	.6269
	Test	**.8801**	.9198	.0922*	.6585*	.6377
GLMnet	**CV**	.7735*	.5993*	.2663*	.4163*	.5139
	Test	.7557	.5974*	.2488*	.3732*	.4938
RF	**CV**	.9021	**.9320**	**.9915**	**.8689**	**.9236**
	Test	.8706	**.9500**	**.9904**	**.9348**	**.9365**

This assumption is validated by the higher accuracy of RF predictions since the internal mechanics of this method are well suited to this scenario: in RF, each individual decision tree operates by considering a randomly sub-sampled feature set and making predictions inside this reduced search space.

In Fig. 8 we can also see the prediction error plots (left) and residual plots with distribution histograms (right) for cardanic stiffness when not using control variables in the model. We can see that all the models with the exception of RF suffer a bit in the zones that are close to the range limits of the target. This effect is more clearly visible in the case of linear models. Notice that the range of the vertical axis in the case of the residual plot for RF is much narrower, thus the residuals are much lower than they seem to be when visually compared to the rest of the methods. These differences in plot ranges have been considered necessary for the sake of visibility in each individual plot. Based on these observations, we would conclude that RF is the adequate algorithm for incorporating new designs to the catalog.

VI. CONCLUSIONS AND FUTURE WORKS

In this paper we have analyzed the suitability of machine learning surrogate models that can be used to speed-up the design of bushing components. Our analysis focuses on light-weight models both in terms of the amount of data, optimized by means of DoE, and the complexity, avoiding too complex models (Occam's razor) that would need a lot of data and computational power, such as deep learning approaches. The main motivation for this is that light-weight surrogate models are more amenable for future on-the-fly hybridization with single and multi-objective design space optimization strategies [23].

The proposed kernel-grounded DoE procedure allowed us to generate the required datasets for each stiffness output in a reasonable amount of time. These datasets were proven to be sufficient for obtaining very good predictions of the outcomes of the FEM simulations using light-weight models, as random forests. Those surrogate models were capable of, not only predicting the impact of enforcing different variations to the geometries of the existing six types of catalog configurations, but also of predicting the performance of potentially new geometries. This fact was checked by training models ignoring

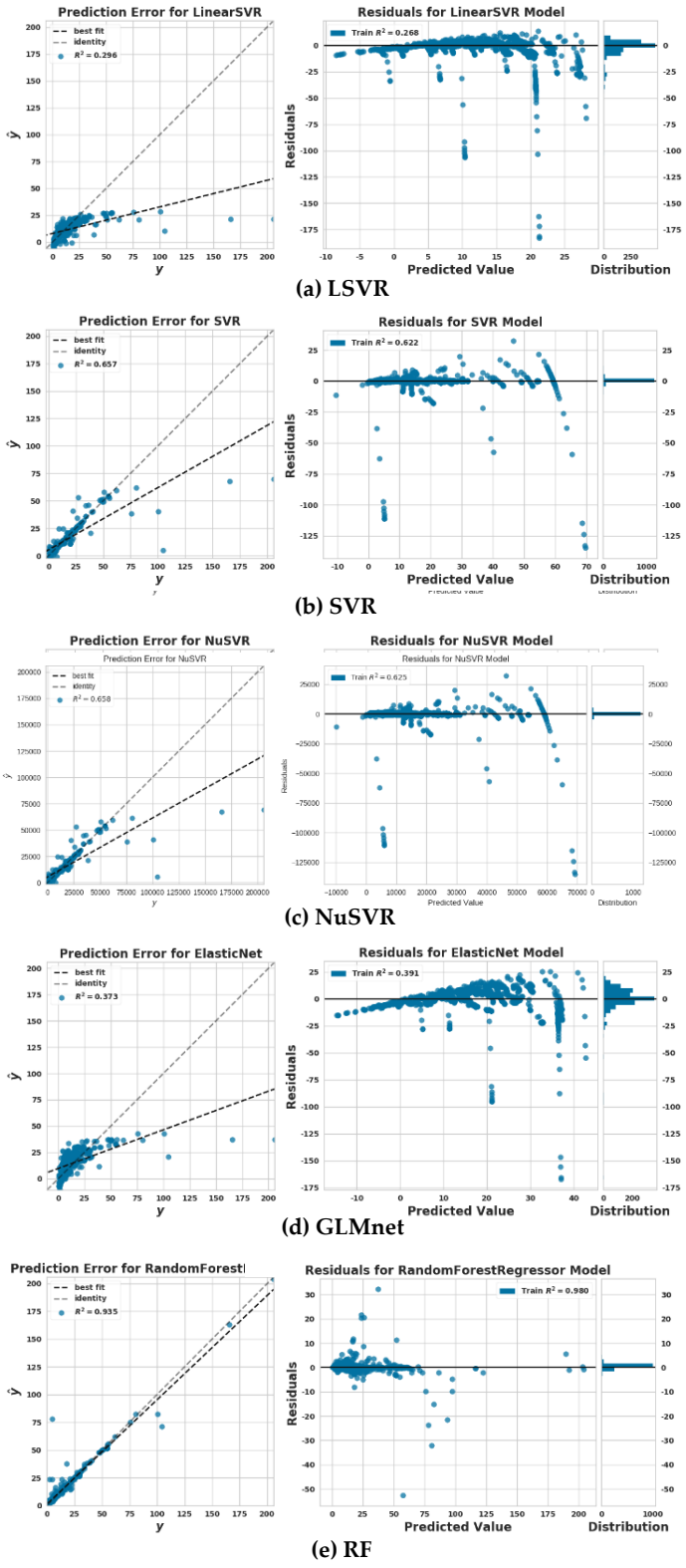

Fig. 8. Prediction error plot (left) and residual plots with distribution histograms (right) for cardanic stiffness prediction without control variables.

the catalog control variables. Nevertheless, this goodness-of-fit should only be extrapolated to configurations in which the control variables are within the ranges of the six configurations in the dataset.

As future research lines, we could try to enhance the DoE by also focusing on general density estimators. This could theoretically allow us to decrease the volume of data without loosing predictive power. Apart from this, the best performing model in our experiments, random forest, is known to be one of the algorithms that is being widely studied by the explainable artificial intelligence community and this is reflected by both novel theoretical contributions [24] [25] and an increasing application to in industrial scenarios [26]. Thus, by further exploring the underlying reasons behind the presently reported superior RF predictions, we could generate wide-interest RF modeling insights.

Finally, if we understand the design process as an optimization problem in which several objectives/requisites have to be optimized in parallel, then multiobjective evolutionary algorithms could be employed. Such algorithms are known to need many objective functions calculations, being FEM simulations included. Surrogate models could be the tool that makes the multiobjective optimization problem solvable in an acceptable time.

ACKNOWLEDGMENT

This work has been partially developed by the Intelligent Systems for Industrial Systems group and the Surface Technologies research group supported by the Department of Education, Language Policy and Culture of the Basque Government. It has been partially funded by FATIGUE project that has received funding from the Department of Economic Development and Infrastructures of the Basque Government under the grant agreement KK-2019/00022.

The contribution of Dr. A.-C. Zăvoianu has been supported by the COMET-K2 "Center for Symbiotic Mechatronics" of the Linz Center of Mechatronics (LCM) funded by the Austrian federal government and the federal state of Upper Austria.

REFERENCES

[1] Zienkiewicz O. C., Taylor R. L., Nithiarasu P., Zhu J.Z. : The finite element method, vol. 3 (1977)

[2] Zabala A., Infante-García D., Giner E., Goel S., Endrino J.L., Llavori I.: On the use of the theory of critical distances with mesh control for fretting fatigue lifetime assessment. Tribology International, vol. 142, 105985 (2020)

[3] Hou G., Wang J., Layton A.: Numerical methods for fluid-structure interaction—a review. Communications in Computational Physics, vol. 12, pp. 337–377 (2012)

[4] Ezkurra M., Esnaola J.A., Martínez-Agirre M., Lertxundi U., Etxeberria U.: Analysis of One-Way and Two-Way FSI Approaches to Characterise the Flow Regime and the Mechanical Behaviour during Closing Manoeuvring Operation of a Butterfly Valve. World Academy of Science, Engineering and Technology, vol. 12, pp. 409–415 (2018)

[5] Lughofer E., Cernuda C., Pratama M.: Generalized flexible Fuzzy inference systems. 12th International Conference on Machine Learning and Applications 2, 1–7 (2013)

[6] Cybenko G.: Approximations by superpositions of sigmoidal functions. Mathematics of Control, Signals, and Systems 2(4), 30–314 (1989)

[7] Hornik K.: Approximation capabilities of multilayer feedforward networks. Neural Networks 4(2), 251–257 (1991)

[8] Wang L.-X.: Fuzzy systems are universal approximators. Proceedings of IEEE International Conference on Fuzzy Systems, 1163–1170, San Diego, CA, USA (1992)

[9] Johrendt J.L., Frise P.R.: Neural Network Bushing Model Development Using Simulation. Proc. 12th Int. Conf. on Advanced Vehicle and Tire Technologies vol.4, 101–109 (2011)

[10] Li L., Sun B., He M., Hua H.: Analysis of the Radial Stiffness of Rubber Bush Used in Dynamic Vibration Absorber Based on Artificial Neural Network. Nueroquantology 16(6), 737–744 (2018)

[11] Huri D., Mankovits T.: Automotive rubber part design using machine learning. IOP Conf. Ser.: Mater. Sci. Eng.659 012022 (2019)

[12] Rodríguez J, Riera F., Plaza J.: Modelling Rubber Bushings Using the Parallel Rheological Framework: Science in the Age of Experience (2017)

[13] Cortes C., Vapnik V.N.: Support-vector networks. Machine Learning 20(3), 273–297 (1995)

[14] Hofmann T., Schölkopf B., Smola A.J.: Kernel methods in machine learning. The Annals of Statistics, vol. 36(3), pp. 1171 –– 1220 (2009)

[15] Schölkopf B., Smola A.J.: Learning with kernels - Support vector machines, regularization, optimization and beyond. MIT Press, London (2002)

[16] Drucker H., Burges C.C., Kaufman L., Smola A.J., Vapnik V.N.: Support Vector Regression Machines. Advances in Neural Information Processing Systems 9, NIPS 1996, 155–161, MIT Press (1997)

[17] Hastie T., Tibshirani R., Friedman J.: The elements of statistical learning: data mining, inference, and prediction (2nd Edition). Springer Series in Statistics, Springer (2009)

[18] Hastie T., Tibshirani R., Friedman J.: Regularized paths for generalized linear models via coordinate descent. Journal of Statistical Software, vol. 33(1), pp. 1 – 22 (2010)

[19] Hastie T., Tibshirani R., Friedman J.: Pathwise coordinate optimization, The Annals of Applied Statistics, vol. 1(2), pp. 302 – 332 (2007)

[20] Breiman L.: Random forests. Machine Learning, vol. 45(1), pp. 5 – 32 (2001)

[21] Pukelsheim F.: The Three Sigma Rule. American Statistician 48, 88–91 (1994)

[22] Wilcoxon F.: Individual comparisons by ranking methods. Biometrics Bulletin 1 (6), 80–83 (1945)

[23] Zăvoianu A.C., Bramerdorfer G., Lughofer E., Silber S., Amrhein W., Klement E.P.: Hybridization of multi-objective evolutionary algorithms and artificial neural networks for optimizing the performance of electrical drives. Engineering Applications of Artificial Intelligence 26(8), 1781–1794 (2013)

[24] Ribeiro M.T., Singh S., Guestrin C.: Why Should I Trust You?: Explaining the Predictions of Any Classifier, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 1135–1144 (2016)

[25] Wei D., Dash S., Gao T., Gunluk O.: Generalized Linear Rule Models, Proceedings of the 36th International Conference on Machine Learning (ICML), 6687–6696 (2019)

[26] Serradilla O., Zugasti E., Cernuda C., Aranburu A., Ramirez de Okariz J., Zurutuza U.: Interpreting Remaining Useful Life estimations combining Explainable Artificial Intelligence and domain knowledge in industrial machinery. Proceedings of IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Glasgow, UK (2020)