Key Engineering Materials
ISSN: 1662-9795, Vol. 926, pp 2193-2200
doi:10.4028/p-zr575d

Submitted: 2021-12-09
Revised: 2022-02-09
Accepted: 2022-02-10
Online: 2022-07-22

Material Parameter Identification of Elastoplastic Constitutive Models Using Machine Learning Approaches

N. Bastos[a], P. Prates[b] and A. Andrade-Campos[c]

Department of Mechanical Engineering, Centre for Mechanical Technology and Automation
University of Aveiro, Portugal

[a]kevinbastos@ua.pt, [b]prates@ua.pt, [c]gilac@ua.pt

Keywords: Material Constitutive Model; Elastoplasticity; Machine Learning; Parameter Identification.

Abstract. Today, the vast majority of design tasks are based on simulation tools. However, the success of the simulation depends on the accurate identification of the constitutive parameters of materials, i.e., its calibration. The classical parameter identification strategy, which relies on homogeneous tests, does not provide accurate and robust results required by the automotive and aerospace industry. Recently, numerical inverse methods, such as the Finite Element Model Updating and the Virtual Fields Method, have been developed for identifying constitutive parameters based on heterogeneous tests. Although these methods have proven effective for linear and non-linear models, the parameter identification process is complex, making it computationally expensive. In this work, a machine learning (ML) algorithm is used to pursue the goal of parameter identification of non-linear models using heterogeneous tests. For that purpose, a ML inverse model is trained using the Finite Element model as data source. A statistical analysis is conducted to identify the correlation between the training dataset size, mechanical tests results and the material parameters. The goal is to understand the importance of the different inputs and to reduce the computational time.

Introduction

Sheet metal forming is one of the most important manufacturing processes to obtain high-performance metal parts for industrial sectors such as automotive and aerospace [1]. These industries heavily rely on numerical simulation tools, and the success of these simulations depends on the precision of the input data. So, these industries require extensive knowledge of the material's behaviour, particularly on the choice of appropriate constitutive laws and the accurate calibration of its parameters. The inaccuracy of material parameters may conduct to faulty results.

The classical characterization process establishes the use of a whole range of standard mechanical tests composed of tensile, compression, or shear tests. However, for non-linear constitutive models, there can be many parameters to be identified, resulting in a high number of classical tests needed [1]. For that reason, several attempts have been made to improve this time-consuming and expensive process [2-4].

Nowadays, the most popular approach to calibrate material model parameters is the use of full-field measurement techniques, such as the Digital Image Correlation (DIC). The main advantage of these techniques is their capability of exploring heterogeneous strain/displacement fields. Recently, a number of numerical inverse methods have emerged to deal with this kind of data. In particular, the ones that received more attention are the Finite Element Model Updating (FEMU) and the Virtual Fields Method (VFM). Several studies compared different full-field strategies and proved to be effective for both linear and non-linear models. However, the computational time and a complex implementation process are still huge problems [4,5].

This work aims to develop a machine learning (ML) model using synthetic data from Finite Element Analysis (FEA) simulations to solve the inverse problem. In the inverse problem, the objective is to identify the unknown material parameters knowing the geometry, boundary conditions, displacements, and strains. Figure 1 describes the correlation between the developed methodology and the direct-inverse problem.

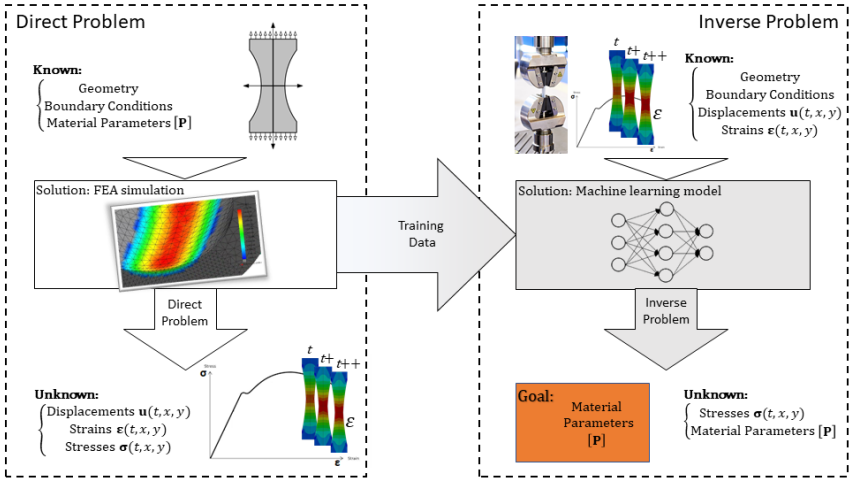


Figure 1 - Correlation between the direct and inverse problem solved by a ML model. The training data is gathered from the direct problem and passed to the ML model. Then, the trained ML model tries to predict the material parameters from experimental data (inverse problem).

Methodology and Implementation

General Approach. Figure 2 represents the general methodology of this work. As stated before, the main objective of this work is to identify the unknown constitutive material parameters from the inverse problem. The ML model created is trained with data obtained from the FEA (direct problem) and then predicts the material parameters from the strains and loads. Therefore, synthetic data is generated by FEA simulations with different combinations of constitutive parameters while maintaining the same geometry, load conditions, and displacements. A Python script was created to enhance the whole process and automatize the generation of the database. Each simulation is divided into twenty time-steps equally spaced and the force, the strain ε_{xx}, ε_{yy}, and τ_{xy} of each element is stored in Comma Separated Values (CSV) files, where each file represents a different simulation with different parameters.

After finishing the process, the data is normalized/standardized using the StandardScaler function presented in the scikit-learn library [6]. The dataset is split into training (90%) and testing sets (10%) randomly.

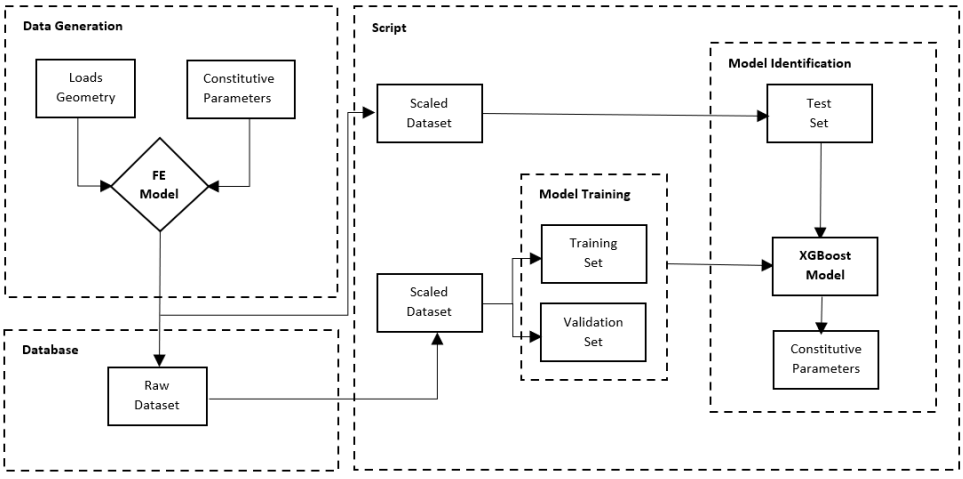


Figure 2 – Flowchart of the proposed identification methodology.

GBoost. Although all previous works used artificial neural networks [7-12] in the identification of material parameters, several studies showed that the XGBoost model outperforms deep neural network models on tabular data [13]. Furthermore, the hyperparameter search process is much faster in XGBoost [13].

Extreme Gradient Boosting (XGBoost) is an efficient open-source implementation of the gradient boosted trees algorithm proposed by Chen and Guestrin [14]. XGBoost is the most used algorithm by the community in regression and classification problems, achieving state-of-the-art results on many tabular datasets [15,16]. Gradient boosting is an algorithm in which new models are created from previous models' residuals (weak models) and then combined to make the final prediction (strong models). When adding new models, it uses a gradient descent algorithm to minimize the loss. XGBoost attempts to minimize the regularized objective as follows:

$$obj(\boldsymbol{\theta}) = \sum_i L\,(\hat{y}_i, y_i) + \sum_k \Omega\,(f_k), \tag{1}$$

where L is the training loss function that measures the deviation between the value $\hat{y}_i$ predicted by the model and the actual value y_i. Ω is the regularization function that measures the complexity of the model, which tends to prevent overfitting. θ represents the set of parameters to be calibrated during training. In order to minimize the objective function, XGBoost uses parameters to find an optimal tree structure employing a greedy search algorithm [14]. Parallel, distributed, out-of-core and cache-aware computing makes the algorithm more than ten times faster than popular models used in machine learning (ML) and deep learning (DL) [13]. Another advantage of this algorithm is that it is well optimized and scalable.

Evaluation Parameters. In this work, prediction performance of the model was evaluated by comparing the predicted and simulated parameters using three measures: the Mean Absolute Error (MAE), the Coefficient of Determination (R^2) and the Mean Absolute Percentage Error (MAPE), given respectively by:

$$MAE = \frac{1}{n}\sum_i^n |\hat{y}_i - y_i|, \tag{2}$$

$$R^2 = 1 - \frac{\sum_i^n (y_i - \hat{y}_i)}{\sum_i^n (y_i - \bar{\bar{y}}_i)}, \tag{3}$$

$$MAPE = \frac{1}{n}\sum_i^n \left|\frac{\hat{y}_i - y_i}{y_i}\right| * 100\%. \tag{4}$$

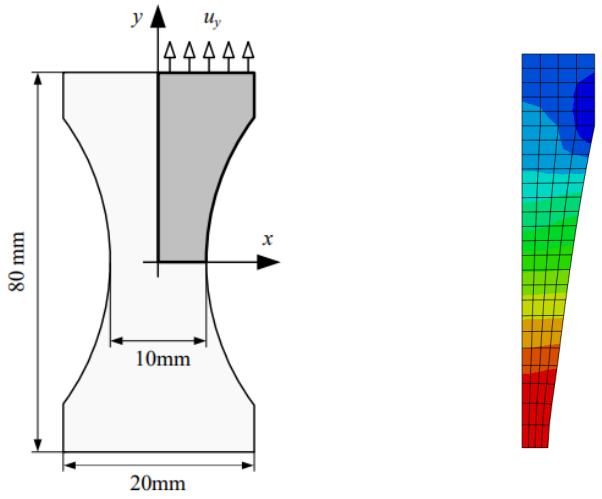
.

Heterogeneous Dogbone Test. All numerical simulations were performed using ABAQUS FEA software. The geometry of the heterogeneous dogbone test specimen is shown in Figure 3(a). This model takes advantage of the symmetries of the problem, which means that only one-fourth of the sample is represented in the FE model (see Figure 3(b)). The mesh is defined as regular with CPS4R (bilinear shape functions and reduced integration), with a total of 100 elements. A boundary condition, with a total displacement of 3 mm along the y direction, is applied to the top boundary of the dogbone (u_y = 3 mm). The constitutive model adopted in this study assumes (i) the isotropic elastic behaviour described by the generalized Hooke's law; (ii) isotropic von Mises yield criterion associated to isotropic hardening described by the Swift's law.

The material parameters related with elastic behaviour, Young's modulus E, and Poisson's ratio ν are assumed as known quantities and take the values presented in Table 1. The input geometry, boundary conditions, load and material parameters are provided from [2]. The material parameters to be determined are K, n and σ_0 given by the Swift's isotropic hardening law:

$$\sigma_y = K\,(\,\varepsilon_0 + \bar{\varepsilon}^p)^n \qquad \text{with} \qquad \varepsilon_0 = \left(\frac{\sigma_0}{K}\right)^{1/n}, \tag{5}$$

where σ_y is the yield stress, K is the hardening coefficient, n is the hardening exponent, σ_0 the initial yield stress and $\bar{\varepsilon}^p$ the equivalent plastic strain.



(a) (b)

Figure 3 – Heterogeneous Dogbone test: (a) geometry, dimensions, and boundary conditions; (b) equivalent plastic strain distribution after a displacement of 3 mm of the FEA model.

For the training data set, the range of the three material parameters used to generate the database is shown in Table 1. In total, 5000 combinations of the parameters were generated randomly using the Latin Hypercube Sampling (LHS) function presented in the pyDOE library [17]. LHS divides the parameter space into bins of equal probability to attain a more even distribution of sample points in the parameter space that would be possible with pure random sampling.

Table 1 – Reference elastic material parameters and input space of isotropic hardening parameters for the dogbone test.

Material parameters	Input values
E [GPa]	210
v	0.3
σ_0 [MPa]	80-300
n	0.1-0.3
K [MPa]	280-700

Results, Analysis, and Discussion

This study conducted hyperparameter tuning using the grid search approach to improve the performance of the model. The XGBoost model has several hyperparameters to be tuned to get the optimal model. Three of those parameters are the maximum depth of the trees, the number of trees, and the learning rate. The GridSearchCV() technique presented in the scikit-learn library [18] was used to process this strategy. GridSearchCV considers all parameter combinations to obtain optimal values of parameters. In this method, all possible combinations of parameter values are evaluated, and the best combination is retained.

After training the model, a dataset size analysis is conducted. Five different training dataset sizes were used (750, 1000, 1500, 2500, 4500). Figure 4 shows the Mean Absolute Percentage Error, R^2, and the time to train the model for the different datasets used. As expected, the results keep improving as the number of training samples grows. Despite this, excellent results are obtained with only 1500 samples (in terms of prediction performance vs computational cost trade-off).

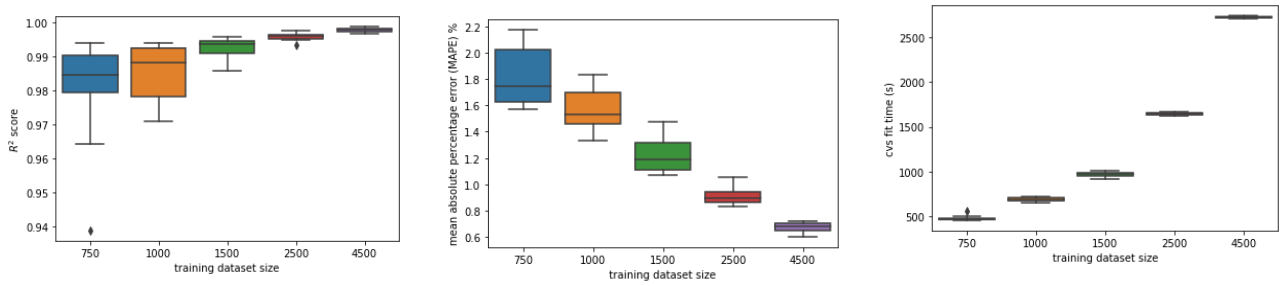


Figure 4 – MAPE, R^2 and training time for the different training dataset sizes.

PCA Analysis. Although the model precisely predicted the parameters, the time necessary to train the model was huge. Therefore, a Principal Component Analysis (PCA) was conducted to improve the computational performance. PCA reduces the dimensionality of the dataset while trying to preserve as much "variability" as possible. Seven different PCA values varying between 0.9 and 0.9999 were used. A 10-fold cross-validation is used to train the model. Figure 5 shows the results of the R^2, MAPE, and time necessary to train the model. Maintaining 0.9999 of the variance, the time to train can be reduced more than 50 times and the model can be reduced to only 72 features while having almost the same R^2 and MAPE score.

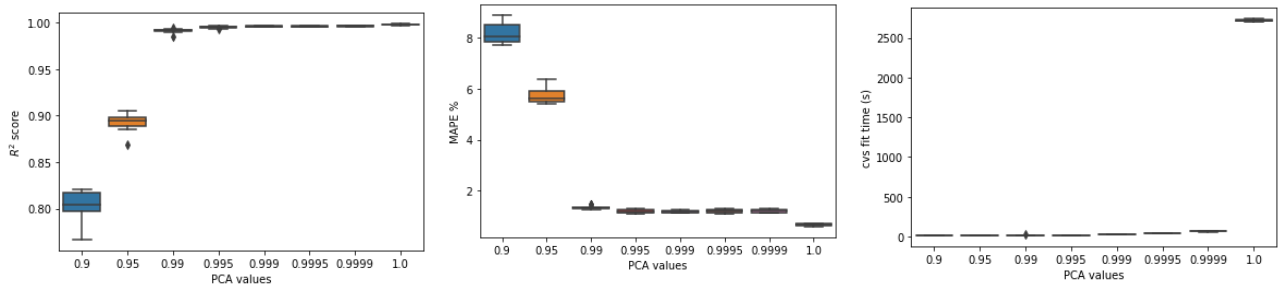


Figure 5 – MAPE, R^2 and training time for the different PCA values.

Model Test. After finishing the PCA process, the model (PCA value = 0.9999) was tested using 450 samples. Each of the 450 samples represent a virtual experiment, synthetically created in the Abaqus FEA. Figure 6 represents the σ_0, n and K values predicted by the model compared with those simulated in Abaqus. The Figure 6 show excellent predictions by the model, not being observed any value significantly different from the expected.

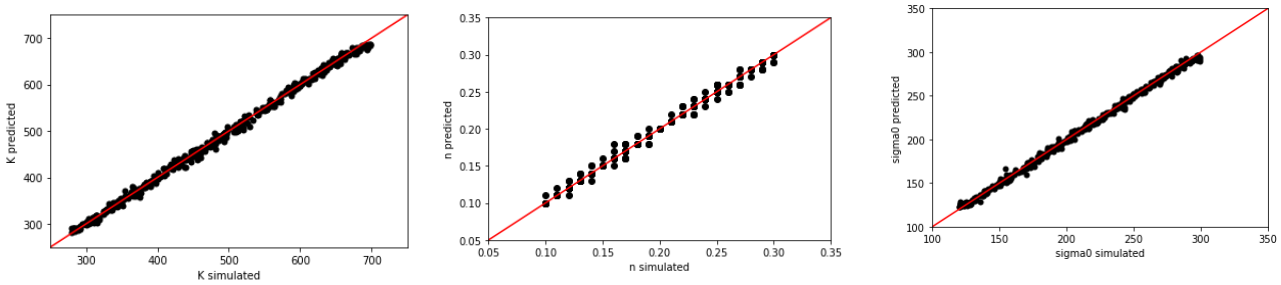


Figure 6 – Predicted vs simulated values for the isotropic hardening parameters.

For further analysis, the three samples with the largest mean absolute percentage error were analyzed; these samples are referred to as “Sim 1”, “Sim 2”, and “Sim 3”. Figure 7 shows the Equivalent Plastic Strain vs Yield Stress for the samples previously mentioned and Table 2 the respective MAPE. This difference is expected since these samples are outside the training data space, and the XGBoost model doesn’t extrapolate. One possible way to improve the results is to train the model with a wider range of parameters.

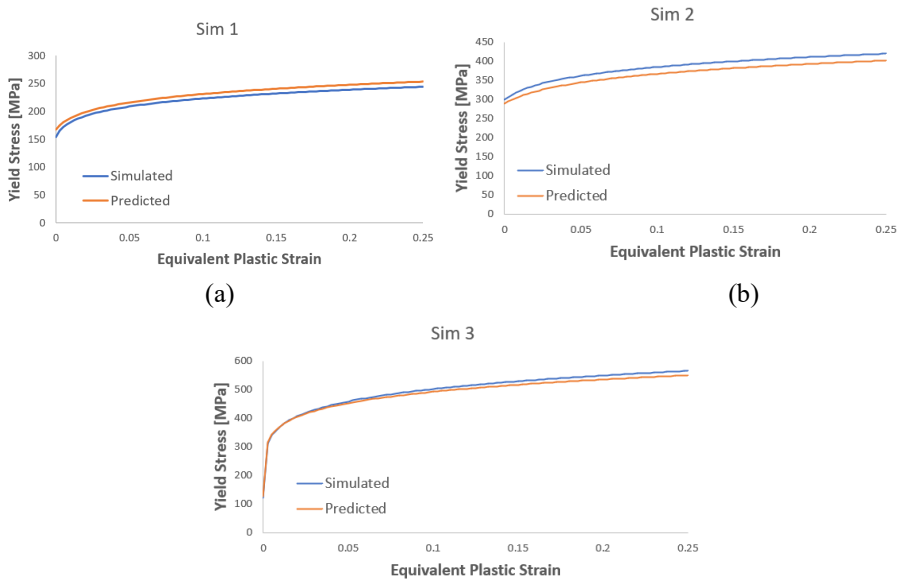
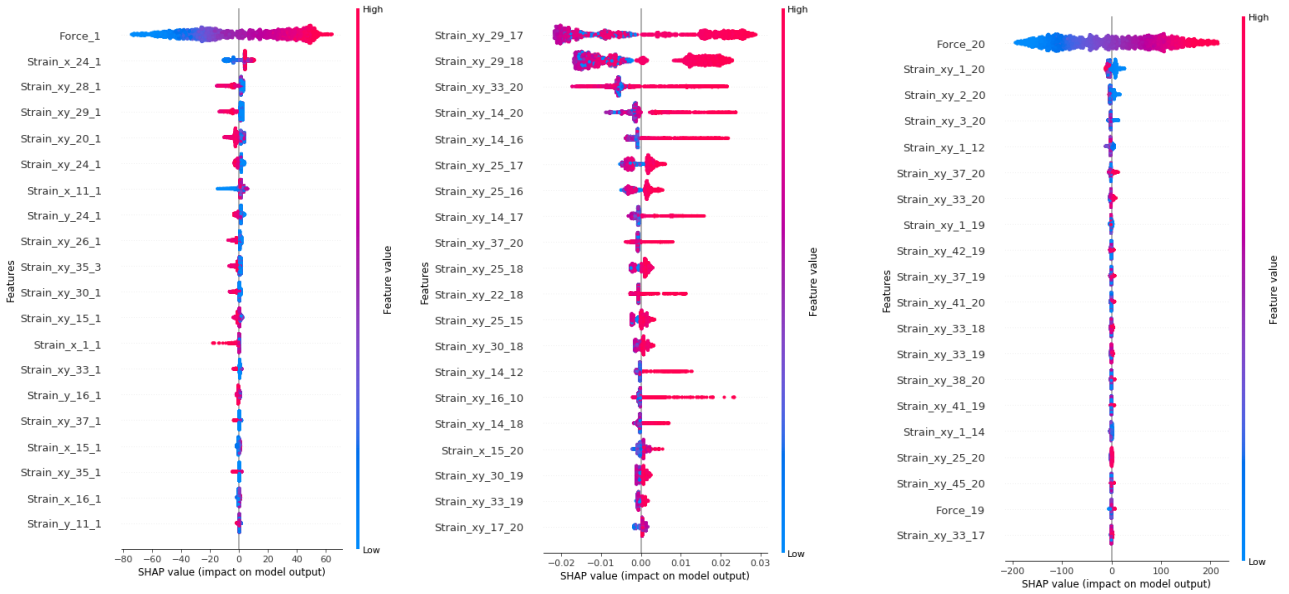


Figure 7 – Predicted vs Reference Equivalent Plastic Strain vs Yield Stress: (a) Sim 1; (b) Sim 2; (c) Sim 3.

Table 2 – Reference and predicted Swift law parameters for the three test samples with largest MAPE values.

		σ_0 [MPa]	n	K [MPa]
Sim 1	Reference	154	0.1	280
	Predicted	167	0.104	292
	MAPE	8.4%	4%	4,2%
Sim 2	Reference	299	0.1	481
	Predicted	289	0.108	460
	MAPE	3.3%	8%	3,3
Sim 3	Reference	122	0.13	678
	Predicted	127	0.121	658
	MAPE	4%	6,9%	3,1%

Feature Importance. As it is hard to gain a comprehensive understanding of their inner work after these have been trained, many ML systems (deep neural networks) are essentially considered black boxes. So, it is really difficult to understand and explain the behaviour of a model. A well-grounded understanding of a model is essential for further improvement and to address its shortcomings. Shapley Additive exPlanations (SHAP) is considered state of the art in Machine Learning explainability [19]. The Shapley value associated with a feature represents the average of its marginal contributions on the predicted values. Figure 8 shows the 20 most important features to predict each parameter: σ_0, n, and K. Each strain feature has a direction (x, y, or xy) and two numbers. The first number represents the element number of Figure 3(b) from bottom left (1) to top right (100), and the last value is the time step number (1-20). The force only has one number that represents the time step. Table 3 shows the error for the 450 test samples used in the previous chapter. The results show that the model only needs five features to have excellent results in predicting the values of the three parameters. This is especially true for σ_0 and K, which depend almost exclusively on the initial and last force, respectively.



(a) (b) (c)

Figure 8 – SHAP values for the different parameters: (a) σ_0; (b) n; (c) K.

Table 3 – R^2 and MAE test score using only the N features.

	σ_0 [MPa]		n		K [MPa]	
N features	R^2	MAE [MPA]	R^2	MAE	R^2	MAE [MPa]
3	0.9974	1.65	0.8765	0.0083	0.9917	8.70
5	0.9994	0.88	0.9716	0.0034	0.9976	4.39
7	0.9995	0.76	0.9710	0.0030	0.9991	2.69
10	0.9995	0.76	0.9688	0.0031	0.9991	2.56
20	0.9995	0.76	0.9888	0.0021	0.9991	2.59

Conclusions

In this work, machine learning models, particularly XGBoost models, are used to pursue the goal of parameter identification of non-linear constitutive models using heterogeneous tests. In this case, the Swift's hardening law parameters were identified with input data generated using FEA simulations. It was concluded:

- Machine learning models can almost perfectly predict the constitutive parameters in heterogeneous full-field mechanical tests without the need of a very large dataset;
- Principal Component Analysis can considerably reduce the computational cost necessary to train the model while maintaining their predictive performance;
- A statistical analysis (SHAP values) to identify the correlation between the material parameters and dataset features can be used to identify the most important features (that may not seem evident) and reduce the computational cost.

Acknowledgments

This project has received funding from the Research Fund for Coal and Steel under grant agreement No 888153. The authors also gratefully acknowledge the financial support of the Portuguese Foundation for Science and Technology (FCT) under the projects CENTRO-01-0145-FEDER-029713, POCI-01-0145-FEDER-031243 and POCI-01-0145-FEDER-030592 by UE/FEDER through the programs CENTRO 2020 and COMPETE 2020, and UIDB/00481/2020 and UIDP/00481/2020-FCT under CENTRO-01-0145-FEDER-022083.

References

[1] N. Souto, “Computational design of a mechanical test for material characterization by inverse analysis," PhD Thesis, PhD Thesis in Mechanical Engineering, Department of Mechanical Engineering, Aveiro University, Aveiro, 2015.

[2] J. Martins, A. Andrade-Campos, and S. Thuillier, “Calibration of anisotropic plasticity models using a biaxial test and the virtual fields method,” International Journal of Solids and Structures, vol. 172, pp. 21-37, 2019.

[3] J.M.P. Martins, A. Andrade-Campos, and S. Thuillier, “Comparison of inverse identification strategies for constitutive mechanical models using full-field measurements,” Int J Mech, vol. 45, pp. 340-345, 1992.

[4] J. Martins, A. Andrade-Campos, and S. Thuillier, “Comparison of inverse identification strategies for constitutive mechanical models using full-field measurements,” International Journal of Mechanical Sciences, vol. 145, pp. 330-345, 2018.

[5] N. Souto, A. Andrade-Campos, and S. Thuillier, “Mechanical design of a heterogeneous test for material parameters identification,” International Journal of Material Forming, vol. 10, no. 3, pp. 353-367, 2017.

[6] J. Martins, S. Thuillier, and A. Andrade-Campos, “Identification of material parameters for plasticity models: A comparative study on the finite element model updating and the virtual fields method,” in AIP Conference Proceedings, vol. 1960,2018.

[7] J. Ghaboussi, J. Garrett Jr, and X. Wu, “Knowledge-based modeling of material behavior with neural networks,” Journal of engineering mechanics, vol. 117, no. 1, pp. 132-153, 1991.

[8] J. Ghaboussi and D. Sidarta, “New nested adaptive neural networks (NANN) for constitutive modeling,” Computers and Geotechnics, vol. 22, no. 1, pp. 29-52, 1998.

[9] A. Javadi, T. Tan, and M. Zhang, “Neural network for constitutive modelling in finite element analysis,” Computer Assisted Mechanics and Engineering Sciences, vol. 10, no. 4, pp. 523-530, 2003.

[10] S. Freitag, W. Graf, and M. Kaliske, “A material description based on recurrent neural networks for fuzzy data and its application within the finite element method," Computers & Structures, vol. 124, pp. 29-37, 2013.

[11] A. Chamekh, H. B. H. Salah, and R. Hambli, “Inverse technique identification of material parameters using finite element and neural network computation," The International Journal of Advanced Manufacturing Technology, vol. 44, no. 1-2, p.173, 2009.

[12] A. Chamekh, H. Belhadjsalah, R. Hambli, and A. Gahbiche, “Inverse identification using the bulge test and artificial neural networks," Journal of Materials processing technology, vol. 177, no. 1-3, pp. 307-310, 2006.

[13] R. Shwartz-Ziv and A. Armon. “Tabular Data: Deep Learning is Not All you Need”, 8th ICML Workshop on Automated Machine Learning (AutoML), 2021.

[14] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794, 2016.

[15] S. Ramraj, N. Uzir, R. Sunil, and S. Banerjee. “Experimenting xgboost algorithm for prediction and classification of different datasets”. International Journal of Control Theory and Applications, pp. 651–662, 2016.

[16] Y. Zhao, G. Chetty, and D. Tran. “Deep learning with xgboost for real estate appraisal.” IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1396–1401. IEEE, 2019.

[17] M. Baudin, “pyDOE”. 2015. https://github.com/tisimst/pyDOE.

[18] scikit learn. “Machine learning in python”. 2008. https://scikitlearn.org/stable/index.html.

[19] S. Lundberg and S. Lee, “A Unified Approach to Interpreting Model Predictions”, 2017.