Key Engineering Materials
ISSN: 1662-9795, Vol. 1049, pp 115-122
doi:10.4028/p-hcCP0o

Submitted: 2026-01-19
Revised: 2026-02-17
Accepted: 2026-03-02
Online: 2026-04-15

Towards DIC-Subset-Independent Machine Learning Models for Constitutive Parameter Identification in Sheet Metal Forming

Dário Mitreiro[1,a*], João Henriques[1,b], Pedro A. Prates[1,c] and António Andrade-Campos[1,d]

[1]Department of Mechanical Engineering, TEMA - Centre for Mechanical Technology and Automation, LASI - Intelligent Systems Associate Laboratory, University of Aveiro, Campus Universitário de Santiago, 3810-193 Aveiro, Portugal

[a*]dario.mitreiro@ua.pt, [b]joaodiogofh@ua.pt, [c]prates@ua.pt, [d]gilac@ua.pt

Keywords: digital image correlation, machine learning, parameter identification, sheet metal forming.

Abstract. Machine learning (ML) algorithms have been studied in literature as an inverse method to predict material constitutive parameters. However, these approaches are often dependent on the mesh discretisation settings applied during numerical simulations, and then difficulty model adaptation to experimental digital image correlation (DIC) subsets. Although a recent study explores the use of an interpolation-based approach to achieve experimental adaptation from numerically-based trained ML models, the proposed methodology lacks evaluation using experimental data. As a follow-up, this study proposes a new evaluation approach. Numerical data is DIC-levelled via MatchID software and then submitted to interpolation. An XGBoost algorithm is then trained on interpolated DIC data and evaluated for parameter prediction, comparing the obtained results with those obtained from the model trained on interpolated numerical data. Overall, the proposed DIC-levelling and interpolation pipeline yields an excellent predictive performance, with results comparable to those obtained when training on interpolated numerical data. The largest deviations are observed for the hardening exponent, while the remaining parameters are predicted with consistently high accuracy. These findings validate the practical applicability of the interpolation-based strategy to reduce the subset scheme dependency of ML models trained on real experimental data.

Introduction

Since the last few years, machine learning (ML) applications in engineering has been constantly increasing due to the substantial increase in computational power and available data. In the field of computational mechanics, the use of ML models as an inverse method for the prediction of material parameters has been a widely explored approach. Although it requires high quantity of available quality data, when compared to other inverse methods, as Finite Element Model Updating (FEMU) or Virtual Fields Method (VFM), ML models show advantages regarding computational time and costs when considering modern complex constitutive models. Besides, instead of relying on analytical models, ML approaches can learn stress-strain relations directly from experimental or numerical data, therefore being a more flexible solution for material parameter prediction [1]. In fact, many recent studies show the successful implementation of artificial neural network (ANN) [2] and extreme gradient boosting (XGBoost) [3] algorithms when applied to parameter prediction in sheet metal forming. However, these methods are often limited to the mesh discretisation scheme applied during numerical stage and, consequently, constrained by the dataset coverage from numerical simulations, which can lead to further difficulties when adapting to experimental digital image correlation (DIC) subsets. To face this challenge, a recent study [4] shows the successful implementation of data interpolation approaches applied to numerical data. However, the mentioned study lacks the demonstration of this approach when applied to real experimental DIC data. Regarding this issue, the present work is a follow-up of the previously mentioned study, by applying the proposed methodology on DIC-levelled data to demonstrate its practical applicability.

Background

Numerical model

The previously mentioned study [4] performs a numerical simulation of a cruciform-shaped specimen, illustrated in Figure 1a, capable of generate heterogeneous stress and strain fields and thus, covering a wide range of scenarios typically found in industrial sheet metal forming processes. The considered orthotropic symmetry and the displacement boundary conditions are represented in Figure 1b. The mesh consists of 564 C3D8R reduced-integration elements, and each simulation (sample) is divided into 20 equally divided time-steps. In each one of the time-steps, forces (F_{xx} and F_{yy}) and strains (ϵ_{xx}, ϵ_{yy}, ϵ_{xy}) for each element are recorded, with a total of 33880 features for each sample. The simulation process is performed in batch mode using Abaqus CAE 2019 software together with its Python 2.7.3 application programming interface (API) [5].

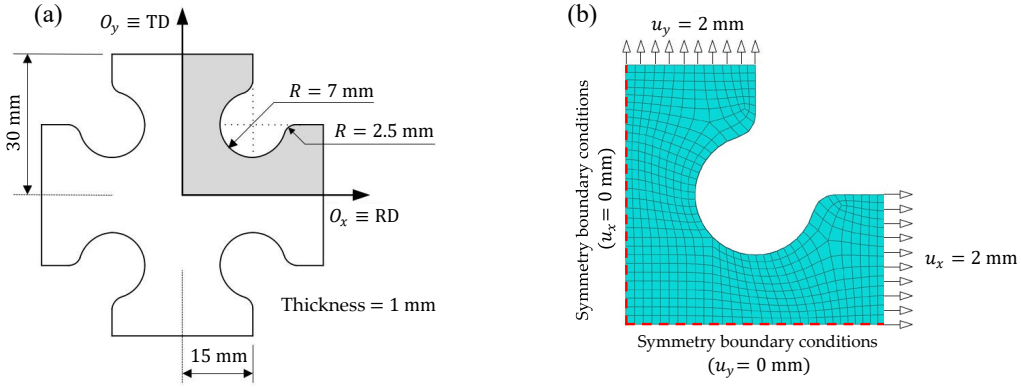

Fig. 1. Biaxial tensile test on a cruciform specimen [4]: (a) geometry and dimensions; (b) boundary conditions and finite element mesh.

The material behaviour is described by isotropic linear elasticity (Hooke's law, E = 210 GPa and ν = 0.3) combined with orthotropic plasticity governed by the Hill'48 yield criterion [6] and isotropic hardening given by the Swift law [7]. The Hill'48 yield function is written as

$$F(\sigma_{yy}-\sigma_{zz})^2 + G(\sigma_{zz}-\sigma_{xx})^2 + H(\sigma_{xx}-\sigma_{yy})^2 + 2L\tau_{xz}^2 + 2M\tau_{xz}^2 + 2N\tau_{xy}^2 = Y^2, \quad (1)$$

where F, G, H, L, M, N are anisotropy coefficients (in this study, L = M = 1.5), σ_{xx}, σ_{yy}, σ_{zz}, τ_{yz}, τ_{xz}, and τ_{xy} are Cauchy stress components in the sheet material frame, and is the current yield stress. By assuming G + H = 1 (i.e., $\sigma_{xx} = Y$), the Hill coefficients can be related to the Lankford ratios r_0, r_{45} and r_{90} as

$$F = \frac{r_0}{r_{90}(r_0+1)}; G = \frac{1}{r_0+1}; H = \frac{r_0}{r_0+1}; N = \frac{1}{2}\frac{(r_0+r_{90})(2r_{45}+1)}{r_{90}(r_0+1)}, \quad (2)$$

Once σ_0, F, G, H, and N are defined, the anisotropy coefficient $r(\alpha)$ and the initial yield stress in tension $\sigma_0(\alpha)$ are obtained from

$$r(\alpha) = \frac{H+(2N-F-G-4H)sin(\alpha)^2cos(\alpha)^2}{Fsin(\alpha)^2+Gcos(\alpha)^2}, \quad (3)$$

and

$$\sigma_0(\alpha) = \sigma_0[Fsin(\alpha)^2 + Gcos(\alpha)^2 + H + (2N-F-G-4H)sin(\alpha)^2cos(\alpha)^2]^{-\frac{1}{2}}, \quad (4)$$

respectively. Finally, the evolution of yield stress during plastic deformation is described by the Swift hardening law,

$$Y = K\left[\left(\frac{\sigma_0}{K}\right)^{\frac{1}{n}} + \bar{\epsilon}^p\right]^n, \quad (5)$$

where $\bar{\epsilon}^p$ is the equivalent plastic strain and σ_0, K and n are material parameters.

Dataset generation

During the simulation, a total of 6755 samples is generated using Latin Hypercubic Sampling (LHS) method applied to σ_0, K, n, r_0, r_{45}, and r_{90} material parameters, with defined input range and step size for each parameter. Then, obtained samples are checked for decreasing load during simulation for the last time-step transition (i.e., $F_{xx,19} > F_{xx,20}$ or $F_{yy,19} > F_{yy,20}$). The 2260 samples presenting no decrease in load during simulation are then randomly shuffled and divided into train and test datasets with 2000 and 260 samples, respectively. The structure of each one of these datasets is compared by two different types of data, a feature matrix (x) and a target matrix (y). The feature matrices (x_train and x_test) represent the simulation outputs per sample, with shape $n_{samples} \times n_{features}$. Similarly, the target matrices (y_train and y_test) represent the corresponding parameters used to obtain each one of the samples, with shape $n_{samples} \times n_{parameters}$. Total sizes of both feature and target matrices (datasets) are represented in Table 1.

Table 1. Feature and target matrix shapes for both training and test sets [4].

Training		Test	
Feature	Target	Feature	Target
2000 × 33880	2000 × 6	260 × 33880	260 × 6

Interpolation approach

Both train and test datasets are interpolated to regular grid points with different densities, choosing grids of 20 × 20, 30 × 30 and 40 × 40 points applied to a quadratic shape and then restricted to the cruciform-shaped specimen domain. Within specimen's domain, 20 × 20, 30 × 30 and 40 × 40 grids origin a total of 253, 564 and 1006 points, respectively. Regarding the interpolation methods, Python's Scipy RBFInterpolator class was used, choosing linear, cubic and multiquadric methods and keeping the remaining function parameters as default. Overall, after each possible combination between the chosen grid sizes and interpolation methods, a total of 9 different interpolated datasets is obtained.

XGBoost training and evaluation

Each interpolated train feature dataset was used as input to train an XGBoost algorithm and, accordingly, the corresponding interpolated feature test dataset was used to evaluate the predictive performance of the generated model (called simple testing). Also, to test the robustness of each model, a cross testing was performed using feature test datasets interpolated with other methods different from the one used to train the chosen ML model. Regarding the XGBoost algorithm training process, the hyperparameters are the same across all the models and are kept as default, except for the learing_rate (= 0.02), max_depth (= 4) and n_estimators (= 1000). After comparing the predicted parameters with the true ones and assessing each model's performance using R^2, MAE and MAPE metrics, the preferred interpolation combination is the 30 × 30 grid with the multiquadric method, due to the balance between performance and computational cost.

Case study

To test the practical application, a synthetic DIC sample was created by choosing a random set of parameters from the test dataset and interpolating their corresponding features to a generated set of 5000 randomly distributed points within the cruciform specimen's domain, thus intending to approximate the typical subset density often found in a DIC-acquired sample. The interpolation is performed using the multiquadric method, resulting in 300040 features for the synthetic sample. To finish the practical demonstration, the DIC synthetic sample is interpolated using the previously

chosen preferred grid and method combination and used as input to test the corresponding ML model for parameter prediction. The predicted parameters are compared with the true ones and with the ones predicted by a ML model trained with the original feature dataset without interpolation (called original model). This comparison is performed by visually representing the effect of these parameters on the evolution of the yield stress during plastic deformation, the initial yield stress in tension as a function of loading angle w.r.t. rolling direction, and anisotropy coefficient as a function of loading angle w.r.t. rolling direction. Overall, the predicted parameters using the synthetic DIC sample closely match the true ones and those obtained by the original model, once again confirming the strong predictive capability of the ML model trained on data using a 30 × 30 grid and multiquadric method, and also demonstrating the effectiveness of the proposed interpolation approach when applied to a denser group of points.

Motivation

In summary, the described study addresses the gap found in literature related to mesh discretisation dependency but is rather incomplete due to the lack of validation with real experimental data. The interpolation approach becomes truly relevant when applied to real DIC datasets, reducing sensitivity to the chosen subset scheme. Regarding this, the present work proposes a DIC-levelling process applied to the original train and test numerical datasets, obtaining DIC-levelled datasets. Next, these datasets are then submitted to interpolation using the chosen grid and method. The interpolated DIC train and test datasets are then used as input to train an XGBoost algorithm and test its predictive performance. Lastly, the predictive performance of both ML models obtained from interpolated numerical and DIC datasets is compared, fully validating the proposed methodology. An overview schematic of this methodology is represented in Figure 2.

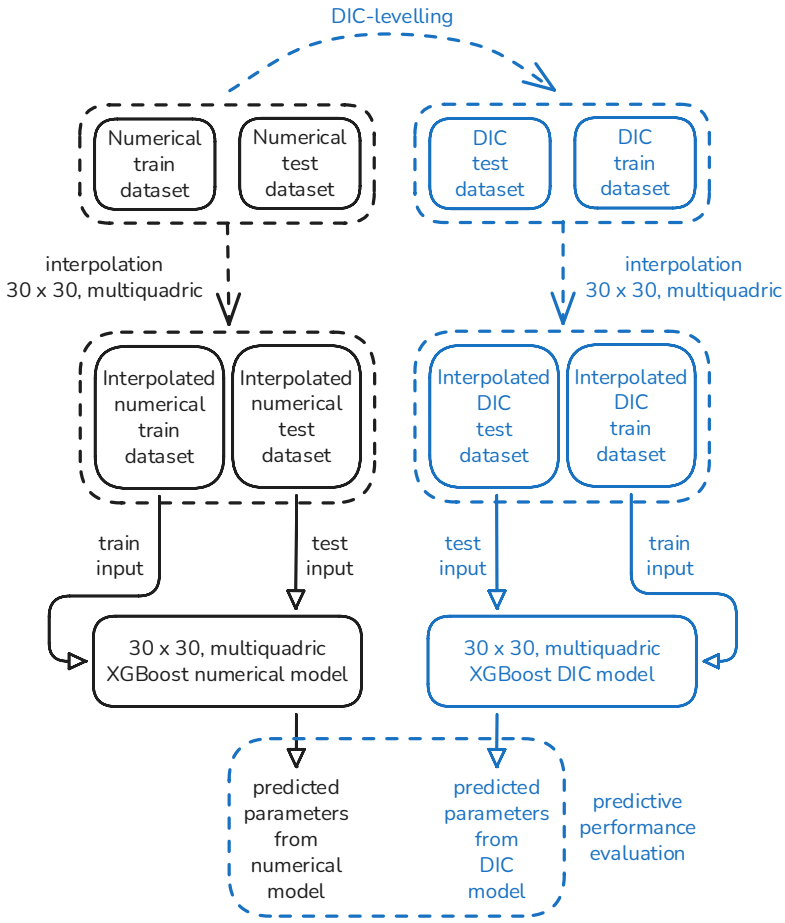

Fig. 2. Schematic for the proposed methodology, represented in blue colour.

DIC Process

The DIC-levelling is applied to the numerical datasets using the MatchID software in stereo (3D) mode, with the settings specified in Table 2.

Table 2. Stereo DIC settings.

Camera	Flir Blackfly BFS-U3-51S5M-C
Focal length	12.5 mm
Image resolution	2448 × 2048 px²
Camera noise	0.48% of range
Working distance	182 mm
Stereo angle	20.74°
Speckle pattern	Numerically generated
Avarage speckle size	8 px
Correlation criterion	ZNSSD
Interpolation	Local Bicubic Splines
Subset shape function	Quadratic
Stereo transformation	Quadratic
Subset size	25
Step size	6
Image pre-filtering	Gaussian, 5 px kernel
Strain window size	15
Strain interpolation	Quadratic quadrilateral Q8
Strain convention	Log. Euler-Almansi
Displacement noise-floor	2.471×10^{-3}
Strain noise-floor	1.13×10^{-4}

However, this process requires a different dataset format, not with forces and strains (as it was previously) but instead with node coordinates of each mesh element. To process each sample, MatchID requires a file with the initial (static) coordinates of each node, plus 20 files (one per time-step) with the updated node coordinates along the simulation. Due to this reason, new numerical simulations are performed for the same 2260 samples, obtaining a total of 47460 data files (21 files per sample). Similarly to the previous numerical dataset, simulations are performed in batch mode using Abaqus CAE 2019 software with an average simulation time of about 30 seconds (Intel Core Ultra 9 285, 24 cores, 128 GB of RAM). Then, the DIC-levelling process itself can be divided into 2 steps: the image generation according to the given speckle pattern, and the DIC filtering process applied to those images to obtain the final results. As the DIC-levelling is performed in 3D mode (to ensure accurate strain values), 2 base speckle pattern images are used in a given angle (stereo angle) to generate 21 images per sample according to the node coordinates from the numerical simulation. In the second step, a DIC filter is applied to all the images of each sample, generating a data file per image containing (among other data) the coordinates and strain values for each subset, with a total of 27296 subsets. Keeping the same hardware specifications mentioned before, computational time is quite significant when performing the DIC-levelling process with about 5 minutes per sample, which results in a total of about 8 days for the entire dataset.

Data Processing and Interpolation

In order to keep consistency with the numerical dataset format, these DIC samples are compiled into a single feature dataset with the forces (F_{xx} and F_{yy}, from the numerical simulations) and the subset strain values (from the DIC-levelling) per time-step, with shape 2260 × 1637800 (samples x features). Then, this dataset is divided into train and test ones, ensuring exactly the same samples and order found in both train and test numerical datasets. Also, to ensure a fair comparison between both numerical and DIC shear strain values, the last ones are multiplied by 2, as MatchID adopts the tensorial convention for the shear strain (ϵ_{xy}), contrary to Abaqus, which uses the engineering shear strain by default (γ_{xy}).

The interpolation process uses Python's Scipy RBFInterpolator class (similarly to the previous study) and is performed for both train and test datasets using the 30 × 30 grid with multiquadric method (preferred combination). However, due to the highest density of points (subsets) from the DIC-levelling, some tweaks regarding the RBFInterpolator class are needed, adjusting neighbors (=250) and epsilon (=10) parameters. Otherwise, if neighbors parameter is not specified, all data points are used by default, leading to a huge unnecessary computational time (around 20 minutes per sample). Also, the epsilon (shape parameter) needs to be tuned so there are no outlier values at the edges of the cruciform specimen.

XGBoost Model

Before the XGBoost algorithm training itself, DIC interpolated feature train data is submitted to a normalisation process, saving the resulting scaler file to later normalise the test dataset upon the model's evaluation process. Similarly to the previous study, hyperparameters are kept as default, except for the learing_rate (= 0.02), max_depth (= 4) and n_estimators (= 1000). Overall, the train process takes around 50 minutes to complete. Lastly, the obtained model is evaluated using the DIC interpolated feature test dataset, also submitted to a normalisation process.

Analysis

Table 3 presents the test performance evaluation metrics for both the results obtained by the numerical and DIC models. Comparing the metrics, the results from the DIC model are very similar to those from the numerical model (obtained in the previous study), showing an excellent predictive performance. The deviations between the parameters predicted by the DIC model and the true ones can be observed in Figure 3, where the overall differences in the predicted values and the performance for each individual parameter is presented. Overall, the highest variation is noticeable by the hardening exponent (n), with a value of $R^2 = 0.96$.

Table 3. Performance metrics for parameter predictions using both models trained with interpolated numerical and DIC-levelled data.

Model	R^2	MAE	MAPE
Numerical	0.992	1.209	0.014
DIC	0.990	1.190	0.014

In addition, a random test sample is chosen to compare the strain values obtained for the last time-step, considering the dataset interpolated from numerical data, and the DIC dataset before and after the interpolation process. This comparison is presented in Figure 4, where ϵ_{xx}, ϵ_{yy}, and ϵ_{xy} strain values are visually similar along all the stages, which demonstrates the good development of the proposed approach along the path.

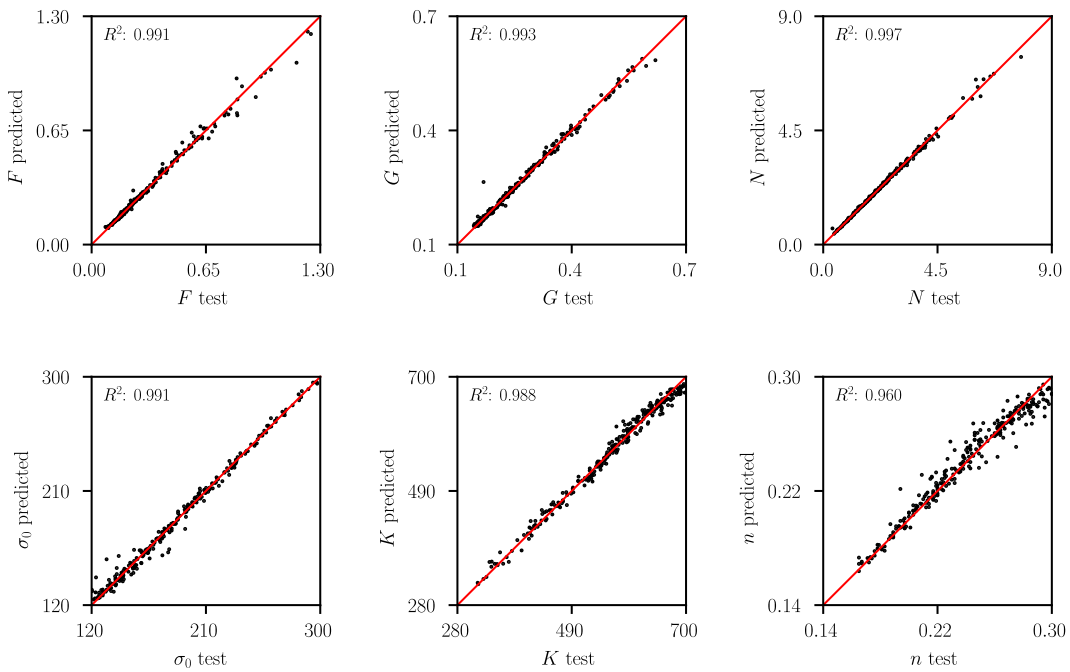

Fig. 3. Comparison between test (true) and predicted parameters from the DIC model.

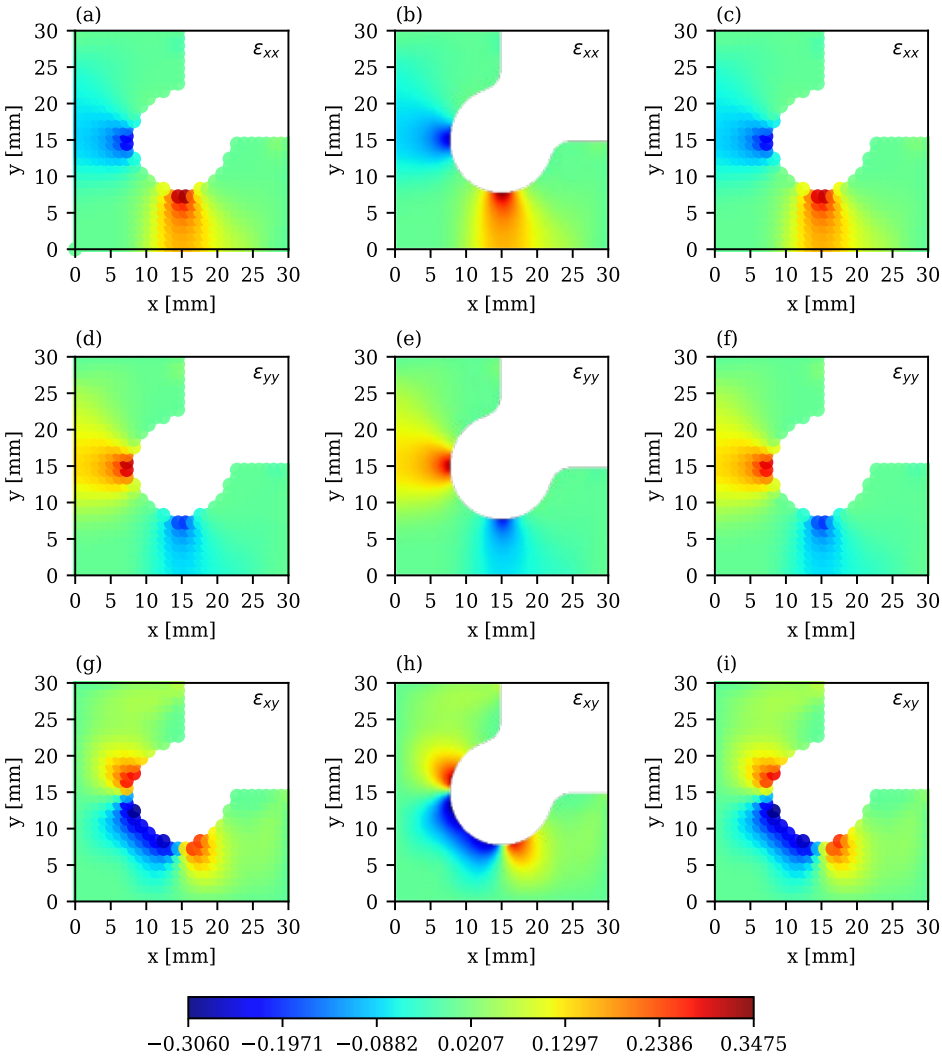

Fig. 4. Comparison between sample strain fields of interpolated numerical data (ϵ_{xx} (a), ϵ_{yy} (d), ϵ_{xy} (g)), DIC-levelled data (ϵ_{xx} (b), ϵ_{yy} (e), ϵ_{xy} (h)) and interpolated DIC data (ϵ_{xx} (c), ϵ_{yy} (f), ϵ_{xy} (i)).

Conclusion

This study proposes a DIC-levelling process, followed by data interpolation, and ML model prediction and evaluation, to assess the practical applicability of a recent explored methodology regarding ML subset dependency reduction. The obtained evaluation results are similar to the previous results using the numerical interpolated dataset and show an excellent predictive performance. When analysing the predicted values for each parameter, the hardening exponent parameter presents the highest variation. Also, the visualisation of strain fields collected from a sample along different stages demonstrate the good development of the proposed approach along the path. These results not only reinforce the efficiency of XGBoost models for parameter prediction using a cruciform-shaped specimen with Hill'48 criterion and Swift hardening law, but also demonstrate the successful application of the proposed interpolation approach, reducing the subset scheme dependency of ML models trained on real experimental data. For future works, a more detailed exploration of alternative interpolation methods and a grid variation study could bring improvements regarding both performance and computational cost.

Acknowledgement

D. Mitreiro is grateful to the Portuguese Foundation for Science and Technology (FCT) for the Ph.D. grant 2025.02607.BD. This work was financed by the Portuguese Foundation for Science and Technology (FCT) and UE/FEDER through COMPETE 2030, of projects COMPETE2030-FEDER-00778700, UID/00481 Centro de Tecnologia Mecânica e Automação (TEMA), and LA/P/0104/2020.

References

[1] P. Böhringer et al., "A strategy to train machine learning material models for finite element simulations on data acquirable from physical experiments," Computer Methods in Applied Mechanics and Engineering, vol. 406, pp. 115894, Mar. 2023, doi: doi.org/10.1016/j.cma.2023.115894.

[2] J. Kim, A. S. Ebrahim, B. L. Kinsey, and J. Ha, "Identification of Yld2000–2d anisotropic yield function parameters from single hole expansion test using machine learning," CIRP Annals, vol. 73, no. 1, pp. 233–236, Jan. 2024, doi: doi.org/10.1016/j.cirp.2024.04.026.

[3] N. Bastos, P. A. Prates, and A. Andrade-Campos, "Material Parameter Identification of Elastoplastic Constitutive Models Using Machine Learning Approaches," Key Engineering Materials, vol. 926, pp. 2193–2200, Jul. 2022, doi: doi.org/10.4028/p-zr575d.

[4] D. Mitreiro, P. A. Prates, and A. Andrade-Campos, "Reducing mesh dependency in dataset generation for machine learning prediction of constitutive parameters in sheet metal forming," Metals, vol. 15, no. 5, pp. 534, 2025. doi: 10.3390/met15050534.

[5] Dassault Systèmes. Abaqus 2016 Analysis User's Guide (Vols. 1–5). Providence, RI: Dassault Systèmes Simulia Corp., 2016.

[6] R. Hill, E. Orowan, "A theory of the yielding and plastic flow of anisotropic metals," Proc. R. Soc. Lond. Ser. A, vol. 193, no. 1033, pp. 281–297, 1948. doi: 10.1098/rspa.1948.0045.

[7] H. W. Swift, "Plastic instability under plane stress," Journal of the Mechanics and Physics of Solids, vol. 1, no. 1, pp. 1–18, Oct. 1952, doi: 10.1016/0022-5096(52)90002-1.