MACHINE LEARNING FOR THE PREDICTION OF EDGE CRACKING IN SHEET METAL FORMING PROCESSES

Armando E. Marques[1], Pedro A. Prates[1,2*], Ana R. Fonseca[3], Marta C. Oliveira[1], Martinho S. Soares[3], José V. Fernandes[1], Bernardete M. Ribeiro[4]

[1]Univ Coimbra, CEMMPRE, Department of Mechanical Engineering, 3030-788 Coimbra, Portugal
[2]Department of Mechanical Engineering, Centre of Mechanical Technology and Automation (TEMA), University of Aveiro, 3810-193 Aveiro, Portugal
[3]Toolpresse, Peças Metálicas por Prensagem, Lda., Parque Industrial De Vendas Novas, 7080-341 Vendas Novas, Portugal
[4]Univ Coimbra, CISUC, Department of Informatics Engineering, 3030-290, Coimbra, Portugal

[*]Corresponding author: prates@ua.pt

Keywords: Sheet metal forming, machine learning, defect prediction, edge cracking

Abstract *This work aims to evaluate the performance of various machine learning algorithms in the prediction of metal forming defects, particularly the occurrence of edge cracking. To this end, seven different single classifiers and two types of ensemble models (majority voting and stacking) were used to make predictions, based on a dataset generated from the results of two types of mechanical tests: the uniaxial tensile test and the hole expansion test. The performance evaluation was based on four metrics: accuracy, recall, precision and F-score, with the F-score being considered the most relevant. The best performances were achieved by the majority voting models. The ROC curve of a majority voting model was also evaluated, in order to confirm the predictive capabilities of the model. Globally, ML algorithms are able to predict the occurrence of edge cracking satisfactorily.*

1. IDENTIFICATION OF THE SECTOR

Sheet metal forming is a manufacturing technique capable of producing a high volume of components at a relatively low cost. Thus, this technique is widely used in the automotive and aerospace industries. The automotive industry is characterised by a constant need for innovation, in order to be able to guarantee profits while ensuring that their products respect quality, safety and environmental impact demands. This industry has great relevance in the European market, and in particularly in Portugal where there are over 400 companies dedicated to the production of components, of which about 8 percent of business volume derives from metallic components.

2. PROBLEM STATEMENT

High strength steels are frequently used in components produced by sheet metal forming processes, particularly in the automotive industry, as their use allows for the reduction of vehicle's mass, while still guarantying good mechanical properties. However, when producing components with these materials, the occurrence of edge cracking is relatively common, particularly when they have bending regions where the ratio between curvature radius and the sheet thickness is low. This edge cracking consists of the occurrence of fractures in a component, usually at the outer edge of a bent area, where the strain path corresponds to uniaxial tension.

3. PREVIOUS SOLUTIONS

The traditional approach to process design was based on trial-and-error. This approach is costly and time consuming, making it unfeasible in today's extremely competitive market. As such, researchers are looking for more efficient alternative methods to apply to process design. At first, the focus was on the application of the finite element method (FEM) for the simulation of forming processes. However, FEM simulations can be computationally expensive when applied to complex forming processes, and a significant number of simulations can be required to obtain good design solutions. Another limitation of the FEM solution is the fact that material variability is not properly taken into consideration. In fact, there can be significant differences in mechanical behaviour between the various metal sheet coils received during production. This difference in mechanical behaviour leads to the occurrence of defects when using some of these material coils during production, which was not predictable during the design process.

4. DESCRIPTION OF THE PROPOSED SOLUTION

In order to reduce the scrap rate associated with the material mechanical behaviour variability mentioned previously, some form of material evaluation should be performed on newly received materials, before production, to confirm whether or not each material has the mechanical properties necessary to avoid the occurrence of defects.

To the author's knowledge, the application of ML algorithms in this type of evaluation is not common, however, these algorithms have proved effective in other applications related to both forming process analysis and material parameter identification. Among the various ML

algorithms available, Artificial Neural Networks (ANN) are the most used. Aghasafari et al. [1] applied ANN to a hot rolling process in order to predict flow stress variations. Many combinations of training algorithms and transfer functions were evaluated in this work, in which the predictive performances obtained by the ANN built with the best combination (Levenberg-Marquardt training algorithm and tan-sigmoidal transfer function) surpassed those obtained by a conventional least-squares method. Hartmann et al. [2] used an ANN to generate tool paths for an incremental sheet metal forming process, using data representing the desired piece geometry as inputs, and the results showed potential for practical applications. Spathopoulos et al. [3] tested the performance of an ANN in the springback prediction of a S-Rail forming process, achieving a promising mean-square error value. Merayo et al. [4] and Koenuma et al. [5] applied ANN for predicting the stress-strain curves of aluminium alloys. The first work focuses on estimating four material parameters, namely, Young's modulus, yield strength, ultimate tensile strength and elongation at break, and then on defining a bilinear approximation of the real stress-strain curve. This approach was based on a relatively large dataset (2000 alloys) and achieves good overall predictions for the material properties, while showing the difficulties that metamodels can have in making predictions for outliers. The second work used crystallographic texture data as input, and considered the stress-strain curve and r-values as outputs. Beskopylny et al. [6] developed an ANN classification algorithm to group various steel grades, in regard to strength features, achieving a 95% predictive accuracy. Masi et al. [7] developed a new ANN approach, called Thermodynamics-based Neural Networks (TANN), consisting on an ANN in which the two basic laws of thermodynamics were encoded directly in the algorithm. This approach guarantees that any prediction remains thermodynamically consistent and proved to be more effective in the prediction of stress, energy and dissipation than standard ANNs. Gorji and Mohr [8] implemented ANN to describe a constitutive material model and applied it to the simulation of tensile tests, obtaining results comparable to those generated by a J_2 plasticity model, which highlights the potential of ANNs to replace conventional models in finite element applications. While it is clear that ANN-based algorithms can achieve very positive results, the capabilities of other algorithms should not be discarded.

In the case of classification problems, alternative classification algorithms have been used in various areas, achieving results that can clearly compete with those of ANNs. Guevara et al. [9] applied ML classification to fog computing applications, including various ANN configurations, decision trees (DT) and support vector machines (SVM). These algorithms were tested for various degrees of noise in the dataset, and DT proved to be the most robust for this application. Cervantes et al. [10] present a study on SVM, highlighting the various applications of these algorithms, including areas like image identification, face detection, written character recognition and some biology applications. Konstantopoulos et al. [11] applied classification algorithms, including ANNs, SVM and random forest (RF), to the identification of relations between the structure and mechanical properties of carbon fibre reinforced polymers. In this case, SVM achieved superior results, particularly for the final validation dataset, although it is worth mentioning that this dataset was small, which led the authors to consider that overfitting may have occurred in the remaining models. Yucalar et al. [12] applied ML classification to the prediction of software defects during the development process. This work focused on the application of ensemble models, including 10 different types of ensemble predictors and 16

different single classifiers as base predictors. This wide variety of analysed classification algorithms highlights the amount of algorithm options potentially available and the need to evaluate these various options to make an informed decision, in terms of which one to use for a certain application.
The current work consists of evaluating the performance of various ML single classification algorithms, as well as of ensemble models, in predicting the occurrence of edge cracking in sheet metal forming processes. The training and performance evaluation processes are based on a dataset generated from experimental results of two types of mechanical tests, the uniaxial tensile test and the hole expansion test, with both showing a strain path of uniaxial tension, similarly to the regions of the component where edge cracking generally occurs.

4.1. Dataset Generation

The success of the learning stage of an ML algorithm depends on the creation of a good dataset. A dataset consists of the collection of all available information about the problem in question, upon which the algorithm training process is based. The dataset used in this work contains experimental results obtained from two different mechanical tests: the uniaxial tensile test and the hole expansion test. These tests were performed for samples obtained from a total of 176 different sheet metal coils, which for the purposes of this work are all considered to be of different materials, which translate to 176 entries in the dataset.
Two uniaxial tensile tests were performed for each coil, one for the rolling direction and the other for an angle of 90º with the rolling direction. The results obtained from these tests are the yield strength (R_e), the ultimate tensile strength (R_m) and the corresponding strain value (ε_{Rm}), and the percent elongation at fracture for initial gauge lengths of 50 mm (E_{50}) and 95 mm (E_{95}). These results correspond to the input variables of the dataset, and the minimum, mean and maximum values obtained for each variable are shown in table 6.1.

	$R_{e,0º}$ [MPa]	$R_{e,90º}$ [MPa]	$R_{m,0º}$ [MPa]	$R_{m,90º}$ [MPa]	$\varepsilon_{Rm,0º}$	$\varepsilon_{Rm,90º}$	$E_{50,0º}$ (%)	$E_{50,90º}$ (%)	$E_{95,0º}$ (%)	$E_{95,90º}$ (%)
Minimum	138.56	137.30	257.10	261.99	0.070	0.008	9.0	11.0	9.54	10.53
Mean	349.31	359.45	424.68	425.74	0.153	0.143	28.70	28.04	23.96	23.35
Maximum	577.93	630.09	615.63	675.04	0.251	0.260	51.0	52.0	40.89	40.65

Table 6.1. Distribution of the input variables

The samples used for the hole expansion tests had an initial hole diameter of 20 mm, and the test were performed up to punch displacement values that guaranteed the fracture of all samples. The result obtained from these tests is the strain at fracture. Its values allow to establish a criterion to define whether edge cracking occurs or not. This criterion is established based on a target strain value, which depends on the component under study. If the strain at the moment of fracture for a specific material is lower than the target strain, then edge cracking will occur for this material. The adoption of this criterion is justified by the fact that a uniaxial tension strain path occurs both at the edge of the hole, during a hole expansion test, and in the critical zones of the components where edge cracking typically occurs. As such, a correlation can be established between the strain values in the critical zones of the component and those achieved

in the hole expansion test. Two different target values were considered in this work. The first is a strain value of 0.725, which corresponds to the average of the strain values obtained for the 176 different materials (these values vary between 0.34 and 0.91, and their distribution is represented in figure 6.1). For this target value, 62 of the 176 materials lead to edge cracking. Since this is a relatively low amount, a second target value of 0.82 was considered, with 123 of the 176 materials leading to edge cracking.

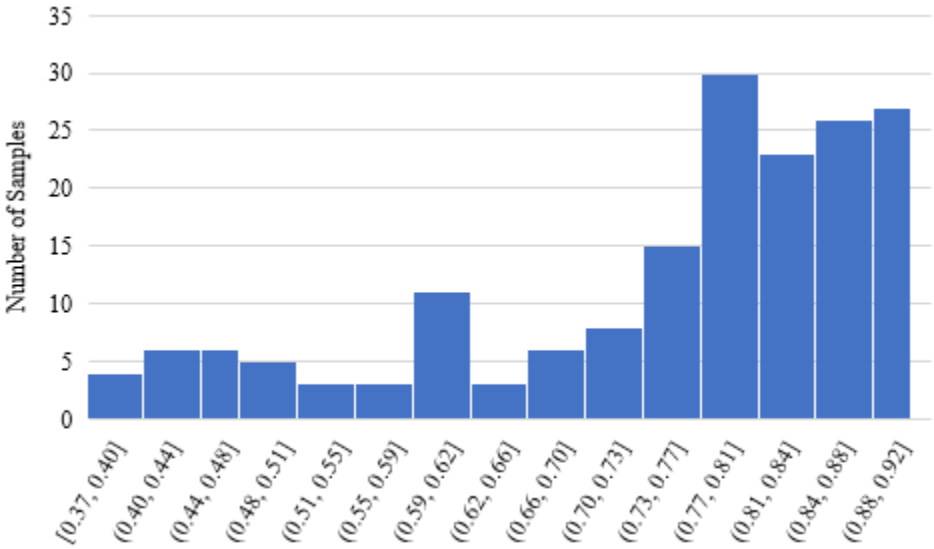

Figure 6.1. Distribution of the strain values at fracture obtained for the hole expansion tests.

4.2. Single classifiers

The single classifier algorithms considered in this work are:

- Multilayer Perceptron (MLP)
- Support Vector Machine (SVM)
- Decision Tree (DT)
- Random Forest (RF)
- Logistic Regression (LR)
- Naïve Bayes (NB)
- k-Nearest Neighbours (kNN)

The models were built using python v2.7.18, with the SciKit-learn library [13]. The theoretical basis of the algorithms is briefly explained in the following sections [14].

4.2.1 Multilayer perceptron

The multilayer perceptron (MLP) is a feed-forward neural network, consisting of one input layer with a number of nodes (neurons) equal to the number of input variables, an arbitrary number of hidden layers (minimum one) and one output layer. Each node is connected to the nodes of the next layer, but there is no connection between the nodes in the same layer. Each

of the nodes of the hidden layers has a nonlinear function called the activation function. The output of a node in a hidden layer or in the output layer is given by the following equation:

$$z_i = \phi\left(\sum_j w_{ij} z'_j + b_i\right), \quad (1)$$

where z_i is the output of the current node i, z'_j is the value obtained from node j of the previous layer, w_{ij} is the weight associated to node i and z'_j, ϕ is the activation function and b_i is the bias term.

4.2.2 Support vector machines (SVM)

The support vector machines (SVM) are discriminative classifiers, which find the optimal separating hyperplane for the training data points, $\boldsymbol{x}$. It consists of identifying the hyperplane that maximizes the distance between itself and any datapoint corresponding to each class. This hyperplane is defined by:

$$(\boldsymbol{w}^T\boldsymbol{x} - b) = 0, \quad (2)$$

where **w** is the normal vector to the hyperplane. Maximizing the distance between the hyperplane and the datapoints of each class means minimizing $\|\mathbf{w}\|$. In order to apply SVM to problems that are not linearly separable, a kernel trick is applied. This consists of transforming the problem space into an implicit, high-dimensional feature space, where linear separation is possible.

4.2.3 Decision tree (DT)

A decision tree (DT) consists of a nonparametric classifier that uses simple rules to continuously split data. These rules are chosen so that information gain is maximized, which means minimizing the following function, F:

$$F = \sum_{i=1}^{m} \frac{n_i}{N_i} H(D_i), \quad (3)$$

where N is the number of examples in the resulting node i and n is the number of examples with the desired label in the mentioned node. H corresponds to an impurity function, such as entropy:

$$H(D) = -\sum_{i=1}^{m} p_i log(p_i), \quad (4)$$

where p is the probability that a given example in the dataset corresponds to label i. The splitting process is repeated until all the samples present in each of the final nodes correspond to the same label, or until an early stopping criterion is satisfied.

4.2.4 Random forest (RF)

The random forest (RF) consists of a combination of several decision trees, which are randomly generated and trained with different parts of the training set. In a classification problem, the predictions from the random forest correspond to the predictions made by more than half of the

decision trees.

4.2.5 Logistic regression (LR)

The logistic regression (LR) consists in fitting a logistic curve, which is a sigmoid curve, to the relationship between the input variables of the training set, x, and the corresponding labels, y. The logistic function is defined as follows:

$$y = \frac{e^x}{1 + e^x} \quad y = \frac{1}{1 + e^{-x}}. \tag{5}$$

This formula can be extended with coefficients α and β, where α is the intercept of y and β is a regression coefficient:

$$y = \frac{e^{(\alpha + \beta x)}}{1 + e^{(\alpha + \beta x)}} \quad y = \frac{1}{1 + e^{-(\alpha + \beta x)}}. \tag{6}$$

4.2.6 Naïve Bayes (NB)

Naïve Bayes is a classifier based on the Bayes theorem, under the (naïve) assumption that every pair of input variables is independent. With the naïve assumption, Bayes' theorem is represented by the following expression:

$$P(y|x_1, \ldots, x_n) = \frac{P(y) \prod_{i=1}^{n} P(x_i|y)}{P(x_1, \ldots, x_n)}. \tag{7}$$

Since, for a given dataset, the denominator will be the same for all entries, a proportionality can be considered:

$$P(y|x_1, \ldots, x_n) \propto P(y) \prod_{i=1}^{n} P(y). \tag{8}$$

The predictions are made based on the label that presents the highest probability:

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y). \tag{9}$$

4.2.7 k-nearest neighbours (kNN)

The *k*-nearest neighbours classifier does not create a model with the training data. Instead, the training data is simply recorded and, each time the model makes a prediction, the distance between each of the training points and the point for which the prediction will be made is calculated. Then, the *k* nearest training points are selected and only these are considered for the prediction. The prediction can simply be the label corresponding to more than half of the training points considered, or weights can be given to each training point, so that from the *k* training points, the nearest ones are more influential.

4.3. Ensemble models

The ensemble models combine various single classifiers, called base learners in this context, to make predictions. In theory, when combining various classifiers, these models are able of

making better predictions, since they have reduced bias when compared with the single classifiers.
Two different types of ensemble models are considered in this work: majority voting and stacking models. In majority voting models, each base learner is trained individually and then is used to make predictions. The class with more than half of the votes is the one predicted by the ensemble model. Stacking models have two levels of learners. First, the base learners are trained and make predictions. Then, a different learner, called a meta-learner, is trained with the predictions of the base learners, and the meta-learner makes the final prediction of the ensemble model.

4.4. Model calibration, training and evaluation

The calibration, training and evaluation processes of the models requires a split of the dataset into a training set, which, as the name suggests, is used to train the models, and a testing set, for which the models make predictions after training. The training of the model consists on adjusting the model parameters, through an optimization algorithm, so that the predictions made by the model for the training set entries match as closely as possible to their real classes.
To enhance the performance of each model, the hyperparameters of each algorithm were calibrated through a trial-and-error process. This is an iterative process, in which a part of the training set is taken out of it to form a validation set. The models are first trained with the reduced training set, make predictions for the validation set and their performance is evaluated; then, hyperparameters are altered to try to improve the results. Once a satisfactory set of hyperparameters is found, the full training set is used in the training of the model and its performance is evaluated for the testing set.
The performance evaluation of the classification models is carried out by means of performance metrics, derived from the confusion matrix. A confusion matrix reports the predictions made by a model with respect to the real classes. For a problem with two classes, the confusion matrix has four elements, as shown in Table 6.2. The performance metrics considered for this work are: accuracy, recall, precision and F-score. Accuracy represents the rate of correct model predictions from the overall predictions. Recall is the rate of positive cases in the testing set that the model correctly identified as positive. Precision is the rate of positive predictions made by the model that are correct. F-score is a more general performance metric that encompasses both recall and precision. It consists of the harmonic mean of these two metrics. The performance metrics are calculated as shown in equations 10-13.

		Real class	
		1	0
Predicted class	1	True Positive (TP)	False Positive (FP)
	0	False Negative (FN)	True Negative (TN)

Table 6.2. Confusion matrix of a problem with two classes

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

$$F - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{13}$$

For the problem addressed in this work, the correct predictions of positive cases (occurrence of edge cracking) are considered a priority and, therefore, more important. Typically, edge cracking occurs for a relatively small portion of the different coils, used during production. This means the dataset is highly unbalanced. This means that a model with low capacity to detect positive cases can still achieve high accuracy, by correctly predicting the negative cases. Thus, accuracy should be considered a support metric, and the performance metrics related to the detection of edge cracking, particularly the F-score, are considered more relevant when evaluating each model.

In order to obtain a more robust performance evaluation, 30 different combinations of training and testing sets were generated randomly from the dataset; for each combination, the training sets contain 70% of the dataset and the testing sets contain the remaining 30%. Performance metrics were calculated for each combination, in order to obtain the mean and standard deviation of each metric, for each ML algorithm.

After the performance of the various models was evaluated, a receiver operating characteristic (ROC) curve was generated for one of the models. This curve consists of a graph that plots the true positive rate (recall) as a function of the false positive rate, calculated as:

$$False\ positive\ rate = \frac{FP}{TP + FP}. \tag{14}$$

The ROC curve is used to evaluate the sensitivity of the model to different thresholds between positive and negative cases, which determine the number of positive and negative cases in the dataset. The generation of this curve required the application of a criterion considering multiple values of target strain, in addition to those considered for the remaining analysis.

4.4. Results and discussion

Figures 6.2 and 6.3 present the performance metrics obtained for the single classifier models. These correspond to the cases with target strain values of 0.725 and 0.82, respectively.

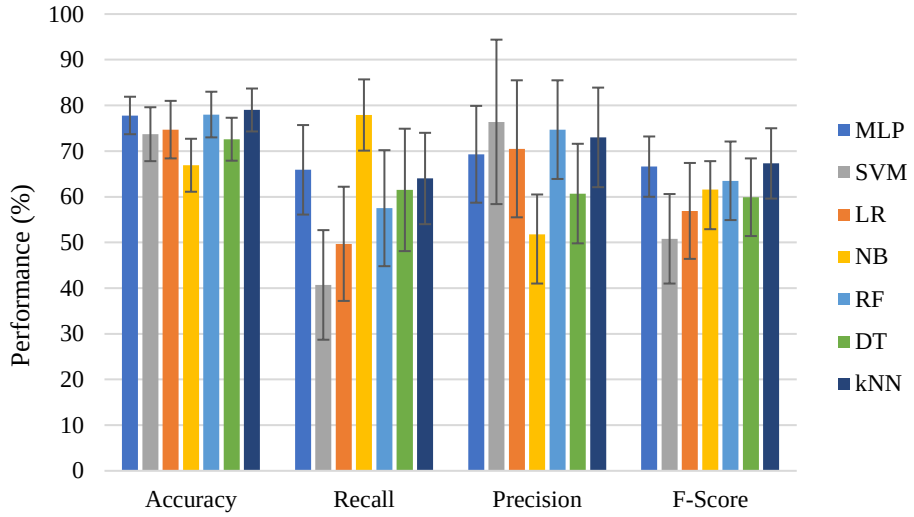

Figure 6.2. Performance metrics for the single classifiers considering a target strain value of 0.725.

For the target strain value of 0.725, accuracy is in general the highest metric, with mean values between 70% and 80% for all classifiers, except NB. The remaining metrics, which focus on the prediction of positive cases, show lower mean values and higher standard deviations. These results can be explained by the low number of positive cases present in the dataset, which means less informative data available on positive cases. The low number of real positive cases also makes each wrong prediction of a positive case have more impact on recall, as it represents a higher percentage of the total positive cases present in the training set. The results obtained for the target strain value of 0.82, which leads to significantly more real positive cases, confirm these conclusions, showing in general better average values and lower standard deviations for recall, precision and F-score, but slightly lower accuracy.

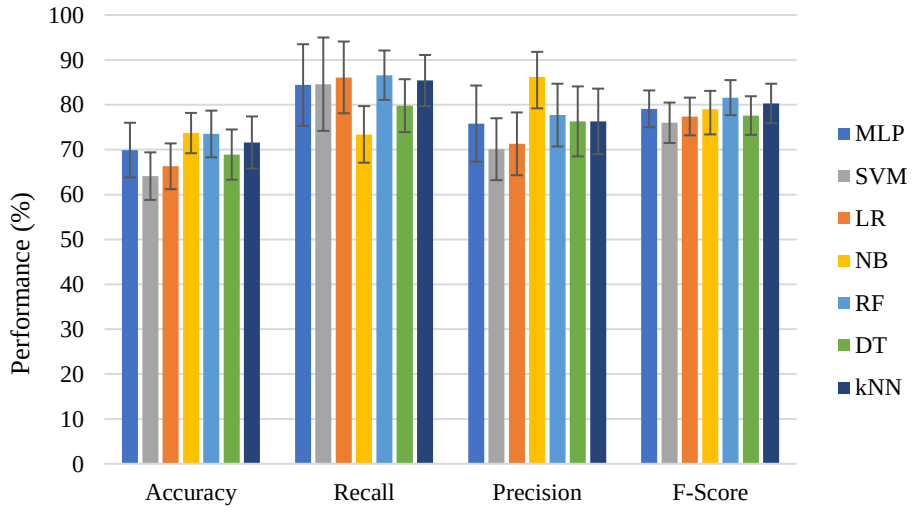

Figure 6.3. Performance metrics for the single classifiers considering a target strain value of 0.82.

There is significantly greater variation between the performance metrics of each classifier for the target strain value of 0.725. Considering the fact that, as previously stated, positive cases (edge cracking) tend to be the exception and not the rule for the problem in question, the target strain value of 0.725 is considered more relevant in determining which classifiers offer the best performances. Thus, based on the F-score values, MLP and kNN single classifiers are considered to have the best performance for this problem with, respectively, 66.6% and 67.3%. The performance metrics obtained for the majority voting ensemble models are shown in Figures 6.4 and 6.5. The labels on the right hand-side of the figures depict which single classifiers are included as base learners in each of the models.

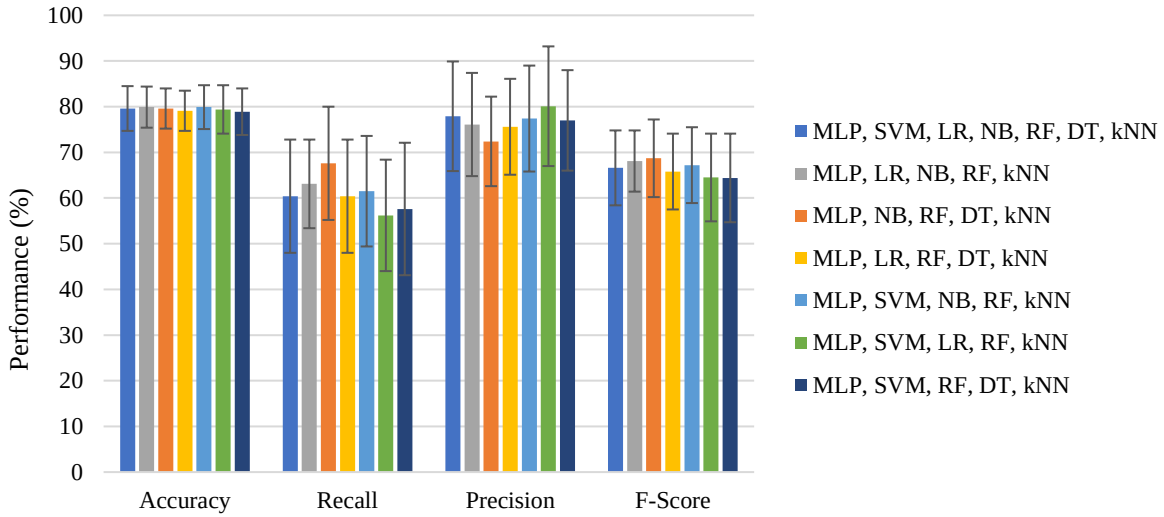

Figure 6.4. Performance metrics for the majority voting models, with a target strain value of 0.725.

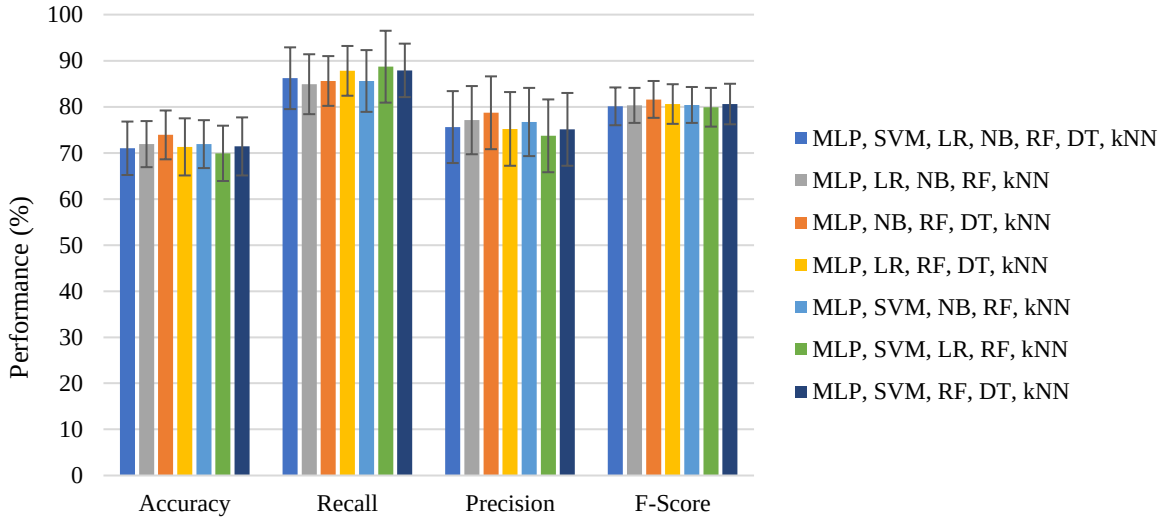

Figure 6.5. Performance metrics for the majority voting models, with a target strain value of 0.82.

When compared with the single classifiers, the majority voting models perform better overall. For the target strain value of 0.725, they have better average values for accuracy (all approximately 80%, which among the single classifiers, only kNN, with 79 %, can compete with) and precision (the highest precision average value achieved by a majority voting model is 80.1%, while the best value obtained by a single classifier, SVM in this case, is 74.7%). The majority voting models with the highest average F-score values also compete with the best-performing single classifiers. The highest average F-Score achieved in this case is 68.7%, which represents an increase of 1.4% when compared to the kNN single classifier. When considering the strain target value of 0.82, the majority voting models achieve mean F-scores of 80%, a value achieved only by the RF and kNN single classifiers, while having slightly superior accuracy, with the highest average value achieved by a majority voting model being 73.9%, while the RF and kNN single classifiers achieved 73.5% and 71.6 %, respectively. It is also worth noting that there is much lower variation in performance metrics between the various majority voting models, when compared to the single classifiers.

Figures 6.6 and 6.7 show the performance metrics obtained for the stacking models. The labels on the right side of the figures identify the single classifier used as meta-learner for each model. The base learners used in each model are the remaining six single classifiers considered in this work. The performance of the stacking models is competitive with the remaining models when considering a target strain value of 0.82, although with slightly higher standard deviations. However, for the target strain value of 0.725, these models perform clearly worse than the majority voting models and the best single classifiers, with lower mean values for all metrics and significantly higher standard deviations. For instance, the average F-score values achieved by these models vary between 58.3% and 63.3%, with standard deviations between 10.8% and 17.1%, while the majority voting models achieve average F-score values between 64.4% and 68.7%, with standard deviations between 6.7% and 9.7%. Thus, the stacking models are not considered effective or reliable for application to the current problem.

Overall, the majority voting models are considered the best choice for application to the problem under analysis, although exceptions can be made for the single classifiers with best performance, since they also present good performances, while requiring significantly less computational power, which can be a relevant factor for cases with much larger datasets. The performance differences between the various majority voting models are relatively small, so the application of any one of them can be recommended.

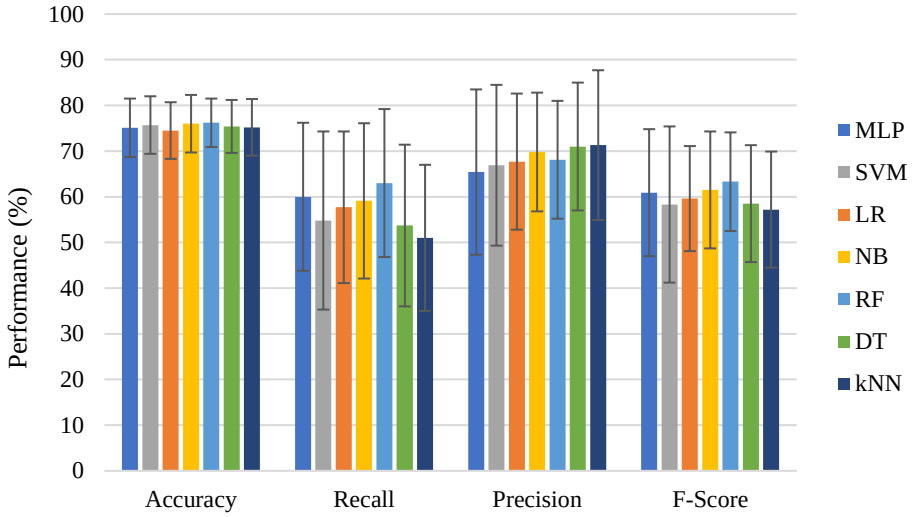

Figure 6.6. Performance metrics for the stacking models considering a target strain value of 0.725.

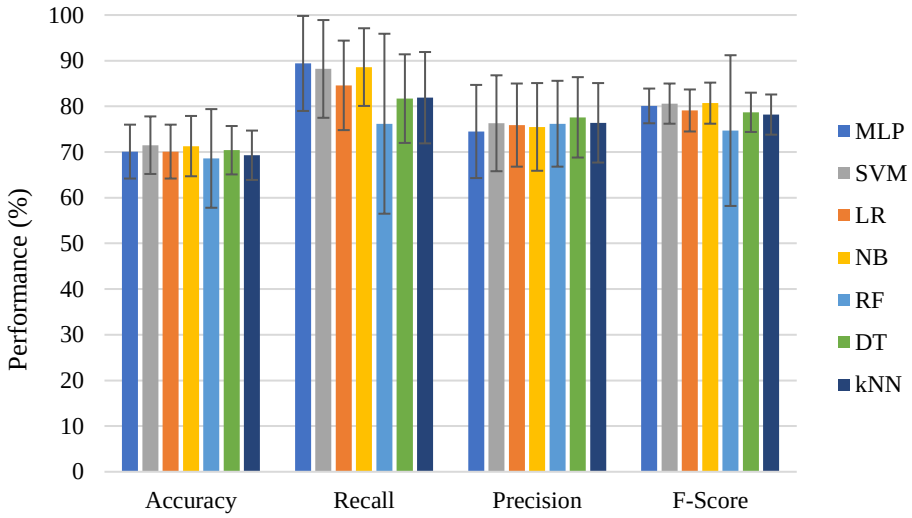

Figure 6.7. Performance metrics for the stacking models considering a target strain value of 0.82.

In order to better understand how the models perform for various different target strain values, a ROC curve was generated for the majority voting model that included all seven single classifiers as base learners. This curve is represented in figure 6.8. Each point of the curve corresponds to a different target strain value, with values ranging between 0.58 and 0.83 being considered. The red line at 45º represents a model that is as effective as a random guess, similar to a coin toss. The ROC curve of a useful classification model will be above this line, and the distance between the two is directly proportional to the quality of the predictions. Another way of relating the ROC curve to the 45º line is through the area under the curve (AUC). The AUC can be used to assess the robustness of a model. A higher AUC means a greater capacity to

distinguish positive from negative cases. The 45º line has an AUC of 50%, while the current ROC curve has an AUC of approximately 85%, which confirms the quality of the model under analysis. The point identified in green corresponds to the target strain value of 0.762, which is close to the median of the strain values at the moment of fracture in the hole expansion test, and represents the best performance of the model among the various target values considered, with a true positive rate of 82% and a false positive rate of 21 %. In practice, this means that the model will achieve better performance the closer the target strain value is to 0.762. It is worth mentioning that this conclusion depends on the dataset. If the dataset is enriched with information from more sheet metal coils, especially if these new coils have significantly different hole expansion strain values at fracture than the current ones, the ideal point would likely change (as well as the ROC curve).

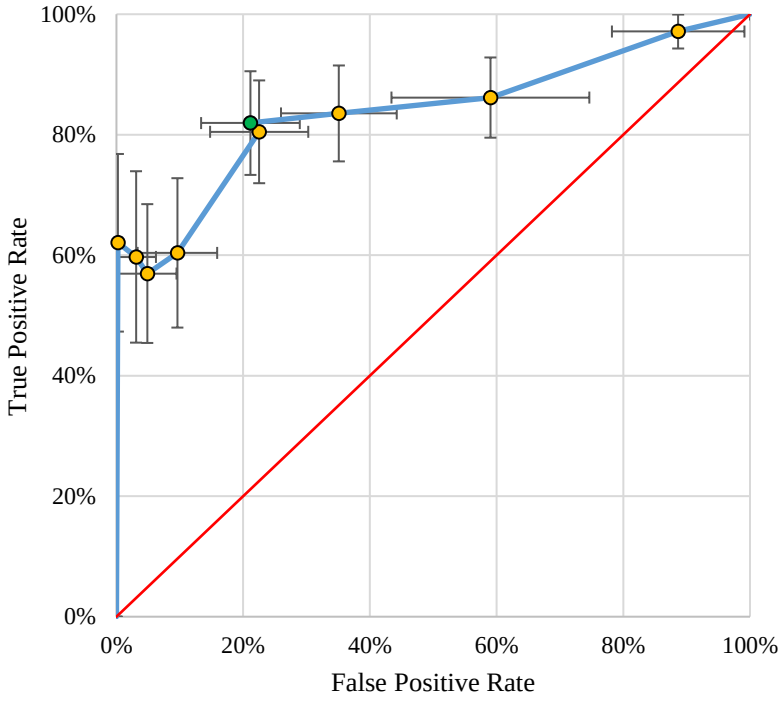

Figure 6.8. ROC curve for the majority voting model that includes all seven single classifiers.

5. CONCLUSIONS AND MAIN ADVANTAGES OF THE PROPOSED SOLUTION

In this work, the performance of various ML algorithms was evaluated in order to predict edge cracking in sheet metal forming. Three stages were considered in this process: first, the generation of the dataset, second, the experimental setup of the models and, finally, the evaluation of the models. Based on the performance metrics carefully chosen for the edge cracking problem, the analysis of the results showed superiority of the majority voting ensemble models over the stacking ensemble or even the single classifiers.

It was concluded that the majority voting models achieve the best performance, with similar

values for the different combinations tested. Among the single classifiers, the MLP and kNN algorithms perform the best, and are valid alternatives to the majority voting models, due to the reduced computational power required. The stacking models, however, show worse performance for low target strain values that the remaining models and, therefore, should not be considered for this type of application.
The analysis of the ROC curve obtained for the majority voting model confirmed the quality of the predictions and shows that the models perform better for target strain values close to the median of the strain values in the dataset. This leads us to conclude that the operational point of the ML models can play a significant role when they are put in practice.
In general, the capabilities of ML algorithms to predict the occurrence of edge cracking are satisfactory, and their application in an industrial environment is recommended. These applications range from the design phase of new components, to the identification of defect probability when using newly received materials, and contribute to the reduction of production costs and scrap rates.

6. ACKNOWLEDGEMENTS

This research is sponsored by FEDER funds through the program COMPETE (Programa Operacional Factores de Competitividade), by national funds through FCT (Fundação para a Ciência e a Tecnologia) under the projects UIDB/00285/2020, UIDB/00326/2020, UIDB/00481/2020 and UIDP/00481/2020; this research is also co-funded by POCI under the projects PTDC/EME-EME/31243/2017 (RDFORMING) and PTDC/EME-EME/31216/2017 (EZ-SHEET) and by the Portuguese National Innovation Agency (ANI) under project POCI-01-0247-FEDER-017762 (SAFEFORMING). All supports are gratefully acknowledged.

REFERENCES

[1] P. Aghasafari, H. Abdi, M. Salimi, Artificial neural network modeling of flow stress in hot rolling, *ISIJ International*. 54 (2014) 872–879. https://doi.org/10.2355/isijinternational.54.872.

[2] C. Hartmann, D. Opritescu, W. Volk, An artificial neural network approach for tool path generation in incremental sheet metal free-forming, *Journal of Intelligent Manufacturing*. 30 (2019) 757–770. https://doi.org/10.1007/s10845-016-1279-x.

[3] S.C. Spathopoulos, G.E. Stavroulakis, Springback Prediction in Sheet Metal Forming, Based on Finite Element Analysis and Artificial Neural Network Approach, *Applied Mechanics*. 1 (2020) 97–110. https://doi.org/10.3390/applmech1020007.

[4] D.M. Fernández, A. Rodríguez-Prieto, A.M. Camacho, Prediction of the bilinear stress-strain curve of aluminum alloys using artificial intelligence and big data, *Metals*. 10 (2020) 1–29. https://doi.org/10.3390/met10070904.

[5] K. Koenuma, A. Yamanaka, I. Watanabe, T. Kuwabara, Estimation of texture-dependent stress-strain curve and r-value of aluminum alloy sheet using deep learning, *Materials Transactions*. 61 (2020) 2276–2283. https://doi.org/10.2320/matertrans.P-M2020853.

[6] A. Beskopylny, A. Lyapin, H. Anysz, B. Meskhi, A. Veremeenko, A. Mozgovoy, Artificial neural networks in classification of steel grades based on non-destructive tests, *Materials*. 13 (2020). https://doi.org/10.3390/ma13112445.

[7] F. Masi, I. Stefanou, P. Vannucci, V. Maffi-Berthier, Thermodynamics-based artificial neural networks for constitutive modeling, *Journal of the Mechanics and Physics of Solids*. 147 (2021). https://doi.org/10.1016/j.jmps.2020.104277.

[8] M.B. Gorji, D. Mohr, Towards neural network models for describing the large deformation behavior of sheet metal, *IOP Conference Series: Materials Science and Engineering*. 651 (2019). https://doi.org/10.1088/1757-899X/651/1/012102.

[9] J.C. Guevara, R.S. Torres, N.L.S. Fonseca, On the classification of fog computing applications: a machine learning perspective, *Journal of Network and Computer Applications*. 159 (2020). https://doi.org/10.1016/j.jnca.2020.102596

[10] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, A. Lopez, A comprehensive survey on support vector machine classification: applications, challenges and trends, *Neurocomputing*. 408 (2020) 189-215. https://doi.org/10.1016/j.neucom.2019.10.118

[11] G. Konstantopoulos, E.P. Koumoulos, C.A. Charitidis, Classification of mechanism of reinforcement in the fiber-matrix interface: application of machine learning on nanoindentation data, *Materials and Design*. 192 (2020). https://doi.org/10.1016/j.matdes.2020.108705

[12] F. Yucalar, A. Ozcift, E. Borandag, D. Kilinc, Multiple-classifiers in software quality engineering: combining predictors to improve software fault prediction ability, *Engineering Science and Technology, an International Journal*. 23 (2020) 938-950. https://doi.org/10.1016/j.jestch.2019.10.005

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, É. Duchesnay, Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research*. 12 (2011) 2825–2830.

[14] M.A. Dib, N.J. Oliveira, A.E. Marques, M.C. Oliveira, J.V. Fernandes, B.M. Ribeiro, P.A. Prates, Single and ensemble classifiers for defect prediction in sheet metal forming under variability, *Neural Computing and Applications*. 32 (2020) 12335-12349. https://doi.org/10.1007/s00521-019-04651-6.