Data-centric and Goal-oriented AI for Robotic Repair Tasks

Kristina Dachtler
Institute AImotion Bavaria
Technische Hochschule Ingolstadt
Ingolstadt, Germany
kristina.dachtler@thi.de

Michael Ortner, Massimo Ferri, Christof Eberst
Convergent Information Technologies GmbH
Haid, Austria
{mortner,mf,ce}@convergent-it.at

Alexander Schiendorfer
Institute AImotion Bavaria
Technische Hochschule Ingolstadt
Ingolstadt, Germany
alexander.schiendorfer@thi.de

***Abstract*—Robotic repair tasks in automotive production, such as spot repair, require heavy physical labor and knowledge about tools and products in order to make informed decisions about the repair strategy. As of now, they typically involve several manual steps even though automation pipelines exist that guide robotic repair using computer vision. Fully automating them requires sensible decision-making based on past data and future expected consequences in terms of costs. We propose to apply machine learning techniques to this problem, show promising first results and discuss challenges for learning systems in this and related manufacturing processes. Using data-centric as opposed to model-centric AI techniques, we were able to improve the overall accuracy by about 6% and using cost-sensitive learning, we were able to better guide model selection towards models that – at the same level in terms of accuracy and F1-score – shift unavoidable misclassifications to less severe outcomes (e.g. falsely treating repairable defects as irreparable).**

***Index Terms*—data-centric AI, machine learning, manufacturing, quality inspection, paint repair**

I. Introduction

Spot repair deals with the repair of paint defects in automotive production. Robotic paint repair is a complex – and still partially manual – task in car manufacturing. The automation of such repair robots presents some major challenges: Not only does the robot arm need to be flexible and application-specific. It also has to detect and locate defects, and then select and execute an appropriate repair program. Automated solutions using machine vision techniques have already been implemented and are in use for defect detection [1], [2]. However, the assessment of the defect and the subsequent selection and implementation of an appropriate repair strategy is often still a manual process. After the automated inspection and detection of defects, they are marked by robots and sent to human workers for repair. This can be very time-consuming and costly, as well as requiring hard physical labor. For this reason, significant time and cost savings can be achieved by automating the entire process chain, including defect identification from cameras using camera vision, feature extraction, repair strategy selection, and finally automated robot programming including planning.

In this paper, we will focus on the selection and implementation of a suitable repair program for an individual defect. Although rule-based techniques, like removal simulations [3] based on 3D-scans of the geometry of the defect located in the CAD model, are theoretically possible, they are limited in this context due to the unique size and shape of each individual defect. It is not feasible to define a universal rule nor cost-effective to derive an exact simulation for processing each specific defect and application scenario optimally.

Instead, we present an alternative approach that uses machine learning (ML) to select a suitable repair strategy based on *data* from previous defect detection and repair steps (see Figure 1). In addition, this approach promises a higher quality and reliability of the selected repair strategies due to its systematic nature, since human decisions are by comparison usually highly subjective. Moreover, it offers the possibility of flexible and, above all, rapid adaptation to changing conditions (e.g. new tools or additional repair programs) or to changing needs of the end user such as, e.g., faster processing times at the possible expense of some wrong repair decisions. Finally, it promises a less time-consuming repair process due to automation, allowing companies to handle a greater number of orders.

Consequently, this paper addresses the following research questions:

- **RQ1**: How well can the data and information of the optical inspection for defect detection in automotive spot-repair be used to automate the selection of the best-suited repair strategy using machine learning?
- **RQ2**: How can expert knowledge be used to improve data quality and therefore model performance?
- **RQ3**: How do we know whether the *data* or the choice of *models* and hyperparameters is the problem, if desired results cannot be reached?
- **RQ4**: How can the best-suited model (in terms of end users' overall application goals) be found and selected?
- **RQ5**: Can a model-based decision provide an advantage in terms of reliability and cost compared to a human decision (which is often based on subjective judgment)?

Our initial hypothesis is that a machine learning model shall be able to predict the desired repair strategy with about 95% accuracy since the use case and data is supposedly easy and manageable compared to other use cases. Moreover, it is routinely solved manually in today's paint shops. Besides

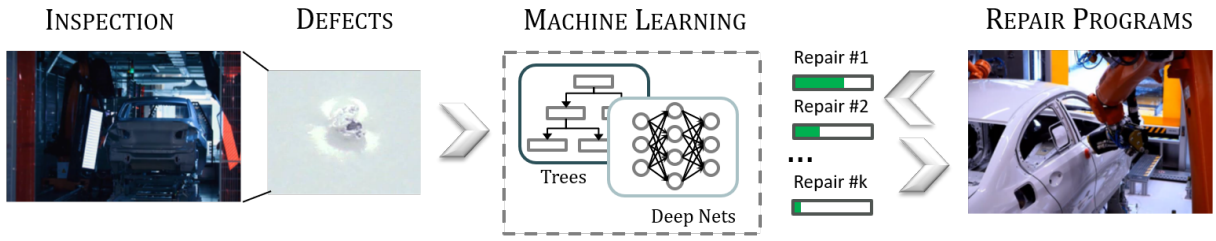

Fig. 1. The core overall approach: A computer vision system feeds in derived features $\boldsymbol{x}$ of a located defect into a machine learning model which learns $p(y \mid \boldsymbol{x}; \theta)$ by adjusting parameters θ. The most likely repair strategy $\hat{y}$ is included in the planning and automatic generation of the robot program.

finding the right repair strategy, one main goal was to isolate defects that cannot be treated at all using the existing repair strategies. The aim of identifying defects that cannot be treated by current repair programs is to prevent wasting resources on unfixable defects, while also ensuring repairable defects are not overlooked or neglected.

In the following,

- We will thus present an ML-based solution and proof of concept for automated spot repair.
- We further show how data-centric methods like the inclusion of expert knowledge for label quality improvement and feature engineering can be used to improve model performance.
- Finally we present proof that optimizing the model performance by common metrics such as accuracy or F1-score does not necessarily meet the main goals of the end user and show a way to find the most appropriate model for a given use case.

II. Related Work

A. Automation in paint (repair) processes

There are several approaches that already integrate machine learning into offline (reactive) robot programming, focusing on similarity-based robot program selection and motion planning support for painting tasks [4], [5]. The decision function for a repair strategy based on a detected defect has not been addressed so far. By contrast, [6] investigates ML approaches based on point clouds that propose ideal robot motion plans and paths for different part geometries. Selecting the repair strategy itself is again not part of the work.

B. Data-centric Machine Learning

The focus in machine learning and artificial intelligence is gradually shifting towards data-centric approaches. Data-centric machine learning, as opposed to model-centric machine learning, refers to improving the performance of models by increasing the quality of the data. The latter focuses on improving performance by optimising hyperparameters or by using different (larger) models and making progress by improving/ changing model designs [7]. By this, models should be developed and good enough to deal with possible noise in the data

The concept of data-centric machine learning involves ensuring good data quality in all parts of the machine learning process. This includes the collection and preparation of training, validation and test data, as well as the subsequent maintenance and assurance of data quality in industrial use. Individual steps include the creation and recording of correct labels for training and testing, feature engineering to increase the information content, or data augmentation methods. Human involvement and participation is, among other things, seen as a crucial factor [7], [8]. Hamid describes data-centric machine learning as a twin driver for compact and robust solutions, with a particular focus on Industry 4.0 [9].

There are several approaches that already prove and show the influence of changes in data-quality on the performance of a model [10], [11]. But all of them test different approaches using different publicly available datasets.

To the best of our knowledge, there is currently no approach that tests and uses data-centric methods to develop a model suitable to be used in industrial practice. Especially not for a manufacturing use case.

C. Cost-sensitive learning

Regarding (supervised) classification tasks, varying misclassifications often result in different costs. Those costs can be included in the decision-making process. The concept of incorporating various costs for diverse misclassifications is not novel and is commonly referred to as *cost-sensitive learning* in the literature. The main idea is to modify the learning process of a model by applying a cost matrix on errors that can be translated into a proper loss function to optimize [12]. This means the costs (if they are known) can already be included at the training algorithm level [13]. As outlined in [12], a cost matrix must be established beforehand and subsequently used for optimizing the model with consideration for costs. Since the costs of different misclassifications are mostly unknown some authors think that this is the reason that this method is not yet widely used [13]. Ideas from optimization might further help to derive numerical costs from qualitative preferences [14].

[15] and [16] already use cost matrices for model optimization in a manufacturing context. In addition, many authors, also in manufacturing, refer to *cost-sensitive learning* when they actually mean reweighting, i.e., increasing the weight of minority and decreasing the weight of majority classes in unbalanced data sets [13], [17]–[20].

In the following, we will use the above-mentioned cost matrix for a goal- and cost-oriented model selection. The cost matrix will not be integrated into the algorithm.

III. Problem Setting, Data and Approach

We first describe the available data and use case, followed by a discussion about the ML engineering steps.

A. Use case and data structure

The ML problem itself is a multiclass classification problem for fault detection and analysis, i.e., given d features $X \subseteq \mathbb{R}^d$ and targets y on a supervised dataset $D = (X, y)$, find an approximation $\hat{f} : X \mapsto y$ using empirical risk minimization on the training set. Alternatively, the problem can be viewed probabilistically in the face of several sources of uncertainty (e.g., label uncertainty and subjectivity, inherent process uncertainty of the sanding/grinding involved) such that for any instance $\boldsymbol{x}$, we learn a discriminative model $p(y \mid \boldsymbol{x})$. The goal is to find the right level of intensity to *repair defects* that resulted from the previous manufacturing step. The initial situation and data structure can be described as follows:

- The targets y consist of 3 classes. Two of those classes (level A and level B) represent two different intensities and strategies for defect repair. In the course of this, level A refers to a "weaker" treatment that takes less time but might not be sufficient for larger defects and level B referring to a "stronger" treatment that reliably resolves most faults at the expense of i) possible material damage and ii) a longer processing time and, thus, costs. The 3rd class, level C, marks all instances with defects that are too severe to be repaired with the existing repair measures. It's important to note that the values for y themselves stem from an imperfect estimation process – levels 3 to 4 would be technically possible but were not part of the data set.
- The features X initially consist of 4 numerical and 2 categorical ones – describing the dimensions and qualitative properties of the faults. These features are derived from an image that itself is not available to us anymore. The latter result from a previous automated optical inspection which is used to detect and mark individual defects.
- The choice of including expert knowledge during feature engineering resulted in two different data sets, as further described in Section IV-A. The number of data points in the first dataset D_{I} is slightly above 3000. The second dataset D_{II} consists of roughly 2500 data points and includes one more numerical feature as a result of the aforementioned discussions.
- In general, the available data was highly imbalanced with on average 82% level A, 16% level B, and 2% level C data points before in the raw data of D_{I} – but this reflects the actually occurring repair strategies.

B. Methodology

As further described in the following sections, we carried out four different experiments. Based on them, we were able to gradually improve our models by sequentially increasing the quality of the data and the amount of information within the data.

We used different algorithms to train and compare the models, reaching from decision tree and random forest classifiers from scikit-learn over gradient boosted trees (`XGBoost`) up to a small feed-forward neural network in Keras/TensorFlow. Since the random forest (RF) and gradient-boosted trees (XGB) consistently showed the best results, in the remainder of this paper, especially the following result section, only those model performances are presented in further detail. To ensure better comparability of algorithms and models, the same models with the same hyperparameters were trained for each experiment. For this purpose, the following combinations were chosen:

a) `'RandomForestClassifier'`:
'min_samples_split' = 2, *'max_leaf_nodes'* = None
b) `'RandomForestClassifier'`:
'min_samples_split' = 10, *'max_leaf_nodes'* = 10
c) `'XGBClassifier'`:
'learning_rate' = 0,05, *'n_estimators'* = 100,
'max_depth' = 10
d) `'XGBClassifier'`:
'learning_rate' = 0,1, *'n_estimators'* = 42,
'max_depth' = 21

In the following they will only be referred to as model *a)* to model *d)*. All presented results were derived on a validation set consisting of 30% of each dataset.

IV. Inclusion of Expert Knowledge for Data Quality Improvement

A. Using expert feedback to increase label quality

Our initial classification results are shown in Table I and referred to as Experiment 1 (E.1). Here, it can be seen that the results are already quite good (considering accuracy), even if they are still below our first target or the expectation of about 95%. However, as mentioned in the introduction, there is another objective to be considered here. The focus should be on *non-repairable defects*, both to avoid trying to fix them (false negatives) and to avoid eliminating repairable defects (false positives). Looking only at the results for class C, it is clear that this objective cannot be achieved. The maximum F1-score obtained with model c) is 14.29%. Models a) and b) don't even recognize a single class C defect. Since this specific problem can be treated as binary also – separating into "repairable" and "irreparable" – we conducted those experiments as well. This resulted in similar model performances, which is why this approach was not pursued any further.

As noted by [7], the subjectivity of labels is often a major challenge when it comes to good data quality. That is why, based on the selected repair strategies, feedback was sought from experts on the extent to which the selected program in y, was suitable for repairing the defect. It describes – in plain text

– if the chosen intensity was deemed good for the respective fault or if a higher intensity, i.e., still more repair work, is needed, or if the fault is not repairable at all. Determining if a weaker treatment would have sufficed is impossible upon seeing the repaired part and thus not part of the dataset.

The process experts' feedback on y, after execution, leads to *adjustments* on the labels based on different rules $\mathcal{R}$.

$$\mathcal{R} = \{(\varphi_i \Rightarrow A_i) \mid i \in I\}$$

where I is the index set of the rules, and $(\varphi_i \Rightarrow A_i)$ is the i-th rule. If the *precondition* φ_i holds for instance $\boldsymbol{x}_j$ the *action* A_i is applied on label y_j, leading to an *adjusted label* y_j^{adj}.

TABLE I
CLASSIFICATION RESULTS WITH DATASET D_{I}. IN E.1, WITHOUT LABEL ADJUSTMENTS, IN E.2 WITH RULE SET $\mathcal{R}_2$

E	Model	Class	Accuracy	Precision	Recall	F1-Score
1	a) (RF)	A	–	97,01%	98,03%	97,52%
		B	–	78,95%	81,82%	80,36%
		C	–	0,00%	0,00%	0,00%
		mean	93,59%	58,65%	59,95%	59,29%
	b) (RF)	A	–	96,36%	98,49%	97,42%
		B	–	81,32%	81,82%	81,57%
		C	–	0,00%	0,00%	0,00%
		mean	93,98%	59,23%	60,10%	59,66%
	c) (XGB)	A	–	97,34%	97,91%	97,63%
		B	–	79,65%	83,03%	81,31%
		C	–	25,00%	10,00%	14,29%
		mean	93,88%	67,33%	63,65%	64,41%
	d) (XGB)	A	–	97,13%	98,37%	97,75%
		B	–	81,70%	81,21%	81,46%
		C	–	20,00%	10,00%	13,33%
		mean	93,98%	66,28%	63,20%	64,18%
2	a) (RF)	A	–	90,72%	97,42%	93,95%
		B	–	57,41%	45,93%	51,03%
		C	–	63,07%	41,84%	50,31%
		mean	85,56%	70,40%	61,73%	65,10%
	b) (RF)	A	–	89,60%	98,52%	93,85%
		B	–	62,38%	46,67%	53,39%
		C	–	87,43%	40,82%	53,69%
		mean	86,42%	76,81%	62,00%	66,98%
	c) (XGB)	A	–	91,00%	96,93%	93,87%
		B	–	55,65%	47,41%	51,20%
		C	–	71,45%	44,90%	53,99%
		mean	85,66%	84,25%	63,08%	66,35%
	d) (XGB)	A	–	90,96%	96,56%	93,68%
		B	–	53,98%	45,19%	49,19%
		C	–	61,43%	43,88%	51,19%
		mean	84,99%	68,79%	61,87%	64,69%

While we could simply perform these adjustments as a fixed preprocessing step, we formalize them explicitly for two reasons: First, in such a manufacturing case, we would keep adapting the labels of past instances based on the results process experts *actually observe* after the predictions have been applied – and thus the data quality is improved for future versions of the models. Second, in different plants different forms of feedback may be used; those adjustment rules have to be kept adaptable.

In the course of our machine learning process, we observed two distinct rule sets which were subject to experimentation, further described as Rule 1 ($\mathcal{R}_1$) and Rule 2 ($\mathcal{R}_2$).

Whenever the feedback states that the chosen intensity was "too low", the level of intensity is increased by one. Whenever the feedback says the defect cannot be repaired, the level is set to C and therefore to "irreparable". An explicit "ok" confirms the label, i.e., $y^{\text{adj}} = y$.

$$\begin{aligned}\mathcal{R}_1 = \{&(\texttt{"too low"} \in \text{feedback} \Rightarrow \lambda y.y + 1),\\ &(\texttt{"irreparable"} \in \text{feedback} \Rightarrow \lambda y.C),\\ &(\texttt{"ok"} \in \text{feedback} \Rightarrow \lambda y.y)\}\end{aligned}$$

By those rules, every data point with intensity level B that was rated as too low, receives an intensity level of C. There is the possibility to add an intensity level B2 to the classification and to the corresponding manufacturing process. However, all of these cases are "synthetical" level B2 cases as this repair program has never been actually performed and evaluated in D_I and D_{II}. Moreover, this leads to an even greater imbalance due to a small number of these cases. Therefore we have not pursued this approach any further.

Instead, a second set of rules was devised. By this set of rules, all data points j with $y_j = B$ that were rated as too low, are converted to level C. Intuitively, if level B is still too low to achieve the desired results, the respective part will be discarded.

$$\begin{aligned}\mathcal{R}_2 = \{&(\texttt{"too low"} \in \text{feedback} \Rightarrow \begin{cases}\lambda y.(y = A) \to B\\ \lambda y.(y > A) \to C\end{cases},\\ &(\texttt{"irreparable"} \in \text{feedback} \Rightarrow \lambda y.C),\\ &(\texttt{"ok"} \in \text{feedback} \Rightarrow \lambda y.y)\}\end{aligned}$$

This means the two experiments shown in Table I can be summarized as follows:

E.1 classification with the initial dataset D_{I}

E.2 classification with labels adjusted by expert feedback based on rule set $\mathcal{R}_2$ in dataset D_{I}

Comparing the results of both experiments directly shows that the accuracy decreased. However, the results of E.1 are somewhat deceiving because the model only reproduced the repair strategy choices before they have actually been validated – therefore training and validation took place on slightly incorrect labels. Moreover, having a closer look on the other metrics shows, that the average precision, recall and F1-score increased for nearly every model. Especially the results for the class C become significantly better due to the fact that now 98 instances fall into class C instead of the previous 20 (cf. Figure 2). A larger part of the irreparable defects can be detected and also the prediction certainty of the class C increases. This can be seen above all in the following confusion matrices, depicted in Figure 2, giving a baseline for **RQ1**.

As can also be seen, this only works at the expense of being able to classify level B defects correctly. But since we were able to make a huge progress in detecting irreparable defects here, we want to continue with the adjusted labels for the following experiments.

B. Nearest neighbor analysis to prove data uncertainties

The results of E.2 shown in Table I are not as good in terms of accuracy as hoped for and could not be improved

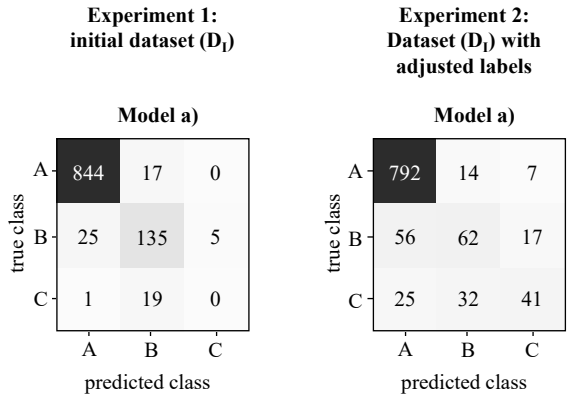

Fig. 2. Confusion Matrix results of experiment 1 and 2 – D_{I} without and with adjusted labels by expert feedback

TABLE II
NEAREST NEIGHBOUR ANALYSIS RESULTS

	Feature 1	Feature 2	Feature 3	Feature 4	Label
Query	10,65	0,71	0,000739	0,008334	**C**
NN	10,68	0,71	0,000969	0,008226	**A**
Index	**Feature 1**	**Feature 2**	**Feature 3**	**Feature 4**	**Label**
Query	22,69	1,76	0,003618	0,01227	**B**
NN	22,43	1,79	0,002389	0,01268	**A**
Index	**Feature 1**	**Feature 2**	**Feature 3**	**Feature 4**	**Label**
Query	3,97	0,47	0,000443	0,001676	**B**
NN	3,95	0,46	0,000278	0,003970	**A**

using hyperparameter optimization. We, therefore, took a data-centric perspective and decided to take a closer look at the individual data points, especially their nearest neighbors (NN).

As can be seen in Table II, using three (slightly cherry-picked) different examples, there are cases where two data points look very similar but do not have the same label. The underlying distribution $p(y \mid \boldsymbol{x})$, thus, shows fairly high (aleatoric) data uncertainty – for some values of $\boldsymbol{x}$, there is no single true answer y. At this point, we assume that the labels used are largely correct. This assumption is valid because the human decisions used as labels so far have already been reviewed and adjusted based on the recorded expert feedback (However, this assumption is challenged in future work as label errors might also be an explanation of varying labels for similar instances, such as those in Table III.) This leads us to conclude that the (few) features available to us may not yet capture all the information necessary to make a reliable decision, i.e., they are simply not predictive enough.

C. Using expert knowledge for feature engineering

Based on the results of the previous section and further discussions, process experts recommended removing one of the categorical features, which according to them should not influence the decision. To avoid possible bias, we removed this feature. This in turn confirmed that assumption as the results did not change in any way.

Furthermore, the process experts recommended one more numerical feature which may contain other, required information. Since it was not possible to obtain those feature values a posteriori to our dataset D_{I}, the second dataset D_{II} was collected. Except for this fifth additional numerical feature and the removed categorical feature, nothing changed in the structure of the data. The results of adding this feature are shown as experiment E.3 in Table III. Since our previous experiments showed better results when using the collected expert feedback for label adjustment, all results from D_{II} are based on previously adjusted labels with the rule set $\mathcal{R}_2$. Compared to the previous experiments E.2, we see an improvement of around 3% in accuracy and up to 6,9% in F1-score (model *a)*), which answers **RQ2**.

One can argue that it is not the same dataset and the increase of performance results is by chance. Therefore experiment E.4 shows the results of dataset D_{II} without using the recommended additional numerical feature. The used data for training and validation as well as the models' hyperparameters are all identical. It can be seen that in this comparison, every model being trained with the additional feature shows better results than without – clear evidence for **RQ3** that feature engineering paid off more than tuning hyperparameters in this case, corroborating the data-centric stance.

TABLE III
CLASSIFICATION RESULTS WITH D_{II} (SIMILAR DATASET LIKE D1 BUT WITH 5TH NUMERICAL FEATURE) IN E.3 AND AS A CONTROL EXPERIMENT WITHOUT THAT FEATURE IN E.4

E	Model	Class	Accuracy	Precision	Recall	F1-Score
3	**a) (RB)**	A	–	93,40%	98,11%	95,70%
		B	–	67,00%	62,04%	64,42%
		C	–	73,68%	45,16%	56,00%
		mean	**89,19%**	78,03%	68,44%	**72,04%**
	b) (RB)	A	–	91,65%	98,58%	94,99%
		B	–	64,29%	58,33%	61,17%
		C	–	91,67%	35,48%	51,16%
		mean	88,32%	83,54%	64,13%	69,11%
	c) (XGB)	A	–	93,51%	97,64%	95,53%
		B	–	64,42%	62,04%	63,21%
		C	–	68,42%	41,94%	52,00%
		mean	88,57%	75,45%	67,20%	70,25%
	d) (XGB)	A	–	93,23%	97,64%	95,38%
		B	–	63,73%	60,19%	61,90%
		C	–	68,42%	41,94%	52,00%
		mean	88,32%	75,13%	66,59%	69,76%
4	**a) (RF)**	A	–	91,98%	97,48%	94,65%
		B	–	60,64%	52,78%	56,44%
		C	–	73,68%	45,16%	56,00%
		mean	87,45%	75,43%	65,14%	69,03%
	b) (RF)	A	–	90,36%	98,90%	94,44%
		B	–	62,07%	50,00%	55,38%
		C	–	86,96%	32,26%	47,06%
		mean	87,21%	79,80%	60,39%	65,36%
	c) (XGB)	A	–	92,32%	96,54%	94,38%
		B	–	59,22%	56,48%	57,82%
		C	–	68,42%	41,94%	52,00%
		mean	86,96%	73,32%	64,98%	68,07%
	d) (XGB)	A	–	92,16%	96,22%	94,14%
		B	–	57,84%	54,63%	56,19%
		C	–	67,50%	43,55%	52,94%
		mean	86,58%	72,50%	64,80%	67,76%

V. Goal-oriented and cost-sensitive model selection

A. Goals for model selection

For the model selection in the specific case study (further described in section III-A), we need to have additional facts in mind. Process experts state that it is of great importance to avoid misclassifications of the irreparable (level C) parts. The consequence of classifying an irreparable part as repairable (*false negatives*) is that there will be the costs of the chosen repair strategy and the resource costs because the part still needs to be disposed of afterwards. Conversely, classifying a repairable part as irreparable (*false positives*) means that resources will be wasted. When in doubt, it is more important for the company to prevent false positives. This has already been shown and observed in previous experiments (experiments 1 and 2).

Considering only the repairable parts, suggesting a level B repair is likely producing a sufficient quality for most cases, even those where a level A treatment would have been enough. But they would be fed back as OK and thus labeled as level B cases for future training sets. Therefore, classifiers need to be incentivized to choose level A in addition to the labels. Potentially we can learn more from choosing level A than level B, which leads to an interesting exploration//exploitation tradeoff. Also, level B is more expensive than level A.

On the other hand, choosing level A first and reworking again (with another round of level A or even level B) will be more expensive than the direct choice of level B. For the company, it is of greater importance to avoid choosing level B if that is not necessary. Level B should only be chosen if it is relatively certain that this intensity is really needed to cure the given defect.

When looking at the performance results from experiment E.3 in Table III, it can be seen that they are very close to each other in terms of both accuracy and F1-score. Model a) "wins" in both metrics, albeit by a small margin and all models are quantitatively similar. However, when looking at the resulting confusion matrices (Figure 3), it can be seen that there is a big *qualitative* difference in *where* the corresponding model errs. For example, model b) exhibits a notably lower error rate in class C (only two false positives) compared to the other models. Therefore the question is which model is the best choice for this use case. In the following, we want to include goals and costs in our decision-making process and see what the cost difference is between these models.

B. Cost-sensitive decision-making

If we evaluate and compare the performance of all models based solely on accuracy, model *a)* would be the obvious choice. When considering the F1-score instead, which is also frequently used for the evaluation of (supervised) classification tasks [16], this would lead to the same result. The accuracy of model *a)* is about 1% better than the other models and F1-score is 2% to 3% better compared to the other models with the same dataset.

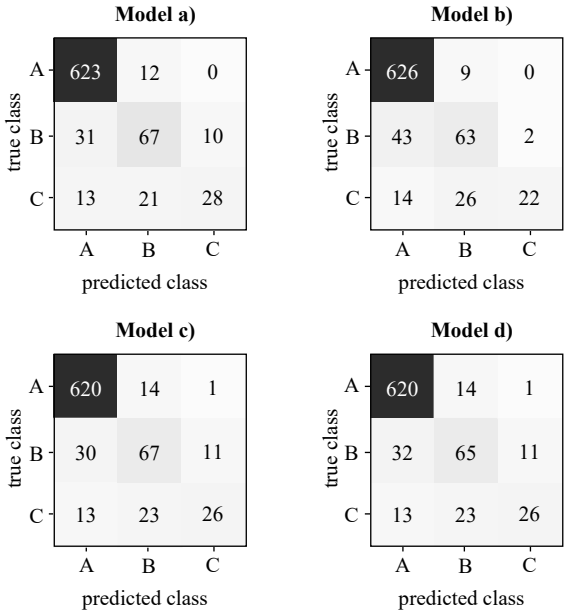

Fig. 3. Confusion Matrix results of E.3 – with D_{II}, adjusted labels by expert feedback $\mathcal{R}_2$ and 5th numerical feature.

Based on the expert statements and main goal of the project, described in the previous subsection already, it is possible to derive (estimated) additional costs for each of the classification results. For example, suppose that the disposal of an irreparable part costs 15 units because of the time and resources involved. The repair of a level B defect costs 3 units and the repair of a level A defect costs 2 units. For all correctly predicted instances $\boldsymbol{x}_i$ there will be no additional costs. For a level B defect, which is considered irreparable, there is an additional cost of 12. The repair of the defect would usually have cost 3. However, the resulting cost is 15 because the part is discarded and resources are wasted. We can summarize those costs in a matrix C_{ij} where *(i,j)* are the costs of an instance of true class j being predicted as class i [12].

This cost matrix for the case study thus looks as follows:

$$C_{ij} = \begin{pmatrix} 0 & 1 & 13 \\ 2 & 0 & 12 \\ 6 & 3 & 0 \end{pmatrix} \tag{1}$$

The sum of the additional costs per instance is given by:

$$J = \sum_{i,j} P_{ij} \cdot C_{ij} \tag{2}$$

Where P_{ij} refers to the confusion matrix resulting from the respective model. For the shown models in experiment E.3, the final, estimated additional costs would be:

a) $J = 335$
b) $J = \mathbf{281}$
c) $J = 366$
d) $J = 370$

Thus, for our use case, **model *b)*** would actually be the most suitable, even though it does not perform best in terms of accuracy, which was *a)*. It even performs worst in terms of F1-score. However, an analysis of the resulting costs reveals that in comparison to model *a)*, model *b)* generates approximately 16% less cost – which better reflects the additional goals expressed by process experts, answering **RQ4**. Clearly, the cost matrices need to be adaptable to different rollouts of the system but the results shows that optimization according to typical metrics is not sufficient.

C. Advantage of a model-based decision compared to a human decision

When investigating **RQ5**, the expert feedback on the previous labels, collected and presented in section IV-A, can be used as a guide to the extent to which a person introduces error into their decision. Human error in this respect is quite normal, as the decision about the severity of a defect and thus the appropriate and optimal repair action is based on subjective, visual perception and judgment. As model-based decision-making promises to improve the quality, reliability, and reproducibility of the selected repair actions, we will explore this in the following in more detail.

As a baseline for the label error rate, the ratio of "not ok" or "irreparable" feedbacks is about 11.9% in D_{I} and about 10.6% in D_{II}. It should be noted that only false negatives can be detected: As already described in section IV-A, feedback can only be given if the selected intensity is not sufficient, i.e. the defect in question would have required a level B if level A had been selected, or cannot be repaired, or a level B defect cannot be processed with the existing repair strategies. It is not possible to see if a misclassification occurs the other way around, i.e. if a lower intensity would have been sufficient, or if a defect marked as irreparable could have been repaired.

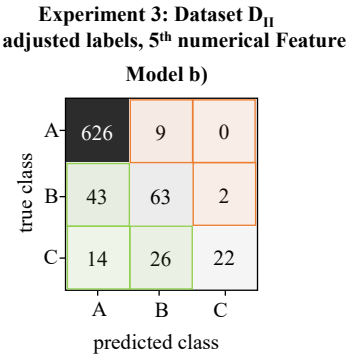

Fig. 4. Comparing model error with human error.

A closer look at these false negatives in our model (marked in green in Figure 4) reveals a total of 83 out of a total of 804 data points in the validation dataset. This is about 10.3%. If the same is calculated from the results of the cross-validation with the training data set, the proportion of false negatives is only around 9.4% which shows that our model is already very good at keeping up with human decisions, if not better.

Assuming that a human decision would make about the same number of errors in the direction of false positives (marked in orange in fig. 4), we can see a clear improvement and, above all, a reproducible and thus reliable decision by the model. In addition to the benefits of automating the spot-repair process, regarding time and cost, the model-based decision making can also demonstrably reduce the cost of misclassification.

VI. Conclusion and Future Work

The automation of repair robots, such as paint repair robots, in automotive manufacturing is a challenging process. An important part of automating the whole process chain from defect detection to defect curation is the subprocess of selecting and executing a suitable repair strategy. As this process is still often carried out manually, there is great potential to reduce time and costs and enable companies to handle higher volumes.

We show a proof of concept on how the above subprocess can be automated by using machine learning. To do so, we use data from the previous subprocess of automated visual inspection for defect detection. Our first model did not yet meet the end users' requirements, so we show how the use of data-centric methods can improve performance step by step. Due to a frequent problem of subjective label assessment, we used expert feedback on our labels to improve our labels and therefore data quality. This step resulted in significantly improved results, especially when considering the classification of faults that cannot be repaired with the existing repair recipes. The average f1-score increased up to 7%. Assuming correct labels, a nearest-neighbor analysis reveals a lack of information in the available features. Through feature engineering, especially collecting and adding one feature based on expert recommendation, we were able to improve model performance a second time. Overall, during the whole machine learning process, we were able to increase the F1-score by up to 12,7% (depending on the algorithm and hyperparameters).

Finally, through goal-oriented model selection using cost-sensitive learning, we were able to select the model that best met the needs of the end users. We used an example to show the impact of choosing a model on the accuracy or F1-score compared to including additional knowledge based on end-user goals and the cost of misclassification to the decision. We can choose a model that has a proven cost advantage over manual human decision-making and can lead to time and cost savings in the overall process.

With an average result of around 88% accuracy, we have not yet reached our goal. Nevertheless, the best model selected is already deployed and used in industrial practice. First investigations show that there is still some uncertainty in the labels. Future research will therefore focus on dealing with these uncertainties – for example by placing them into different levels of confidence for predictions [21]. By using further data-centric methods – e.g. Confident Learning [10] – we plan to gradually improve the quality of the data and thus the performance of the models, to reach the target value.

REFERENCES

[1] Convergent Information Technologies, "Robot paint repair / surface repair with automappps," 17.11.2022. [Online]. Available: https://convergent-it.com/robot-paint-repair/

[2] "Robotic paint repair — 3m abrasives," 24.04.2023. [Online]. Available: https://www.3m.com/3M/en_US/metalworking-us/applications/robotic-abrasives/paint-repair/

[3] G. Coffignal, P. Lorong, and L. Illoul, "A general method to accurately simulate material removal in virtual machining of flexible workpieces," 2015.

[4] D. S. AG, "Artificial intelligence for paint shops," *IST International Surface Technology*, vol. 13, no. 4, pp. 12–13, 2020.

[5] J. R. D. Posada, A. Meissner, G. Hentz, and N. D'Agostino, "Machine learning approaches for offline-programming optimization in robotic painting," in *ISR 2020; 52th International Symposium on Robotics*. VDE, 2020, pp. 1–7.

[6] C. Liu, Z. Yin, and R. Li, "Design of paint repairing robot system based on point cloud," in *2021 13th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*. IEEE, 2021, pp. 162–167.

[7] D. Zha, Z. P. Bhat, K.-H. Lai, F. Yang, Z. Jiang, S. Zhong, and X. Hu, "Data-centric artificial intelligence: A survey," 2023. [Online]. Available: http://arxiv.org/pdf/2303.10158v2

[8] S. E. Whang, Y. Roh, H. Song, and J.-G. Lee, "Data collection and quality challenges in deep learning: a data-centric ai perspective," *The VLDB Journal*, 2023.

[9] O. H. Hamid, "Data-centric and model-centric ai: Twin drivers of compact and robust industry 4.0 solutions," 2023.

[10] C. Northcutt, L. Jiang, and I. Chuang, "Confident learning: Estimating uncertainty in dataset labels," *Journal of Artificial Intelligence Research*, vol. 70, pp. 1373–1411, 2021.

[11] H. Westermann, J. Šavelka, V. R. Walker, K. D. Ashley, and K. Benyekhlef, "Data-centric machine learning: Improving model performance and understanding through dataset analysis," in *Legal Knowledge and Information Systems*, ser. Frontiers in Artificial Intelligence and Applications, E. Schweighofer, Ed. IOS Press, 2021, pp. 54–57.

[12] C. Elkan, "The foundations of cost-sensitive learning," *Proceedings ofthe Seventeenth International Joint Conference on Artificial Intelligence (IJCAI'01)*, 2001.

[13] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert Systems with Applications*, vol. 73, pp. 220–239, 2017.

[14] A. Knapp, A. Schiendorfer, and W. Reif, "Quality over quantity in soft constraints," in *2014 IEEE 26th International Conference on Tools with Artificial Intelligence*. IEEE, 2014, pp. 453–460.

[15] Y. Zhang, Y. Li, Z. Sun, H. Xiong, R. Qin, and C. Li, "Cost-imbalanced hyper parameter learning framework for quality classification," *Journal of Cleaner Production*, vol. 242, p. 118481, 2020.

[16] S. Spiegel, F. Mueller, D. Weismann, and J. Bird, "Cost-sensitive learning for predictive maintenance," 2018. [Online]. Available: http://arxiv.org/pdf/1809.10979v1

[17] G. T. Cinar, J. Thompson, and S. Srinivasan, "Cost-sensitive optimization of automated inspection," in *2015 IEEE International Conference on Big Data (Big Data)*. IEEE, 2015, pp. 1211–1219.

[18] A. Kim, K. Oh, J.-Y. Jung, and B. Kim, "Imbalanced classification of manufacturing quality conditions using cost-sensitive decision tree ensembles," *International Journal of Computer Integrated Manufacturing*, vol. 31, no. 8, pp. 701–717, 2018.

[19] H. Yu, C. Sun, X. Yang, S. Zheng, Q. Wang, and X. Xi, "Lw-elm: A fast and flexible cost-sensitive learning framework for classifying imbalanced data," *IEEE Access*, vol. 6, pp. 28 488–28 500, 2018.

[20] Y. Geng and X. Luo, "Cost-sensitive convolutional neural networks for imbalanced time series classification," *Intelligent Data Analysis*, vol. 23, no. 2, pp. 357–370, 2019.

[21] L. Lodes and A. Schiendorfer, "Certainty groups: A practical approach to distinguish confidence levels in neural networks," in *PHM Society European Conference*, vol. 7, no. 1, 2022, pp. 294–305.