Bauer, Johannes C.; Trattnig, Stephan; Vieltorf, Fabian; Daub, Rüdiger

Handling data drift in deep learning-based quality monitoring: evaluating calibration methods using the example of friction stir welding

Journal of Intelligent Manufacturing

Provided in Cooperation with:
Springer Nature

Suggested Citation: Bauer, Johannes C.; Trattnig, Stephan; Vieltorf, Fabian; Daub, Rüdiger (2025) : Handling data drift in deep learning-based quality monitoring: evaluating calibration methods using the example of friction stir welding, Journal of Intelligent Manufacturing, ISSN 1572-8145, Springer US, New York, NY, Vol. 37, Iss. 2, pp. 759-774, https://doi.org/10.1007/s10845-025-02569-6

Journal of Intelligent Manufacturing (2026) 37:759–774
https://doi.org/10.1007/s10845-025-02569-6

Handling data drift in deep learning-based quality monitoring: evaluating calibration methods using the example of friction stir welding

Johannes C. Bauer[1] · **Stephan Trattnig**[1] · **Fabian Vieltorf**[1] · **Rüdiger Daub**[1,2]

Received: 14 March 2024 / Accepted: 9 January 2025 / Published online: 30 January 2025

Abstract
Deep learning-based classification models show high potential for automating optical quality monitoring tasks. However, their performance strongly depends on the availability of comprehensive training datasets. If changes in the manufacturing process or the environment lead to defect patterns not represented by the training data, also called data drift, a model's performance can significantly decrease. Unfortunately, assessing the reliability of model predictions usually requires high manual labeling efforts to generate annotated test data. Therefore, this study investigates the potential of intrinsic confidence calibration approaches (i.e., last-layer dropout, correctness ranking loss, and weight-averaged sharpness-aware minimization (WASAM)) for automatically detecting false model predictions based on these confidence scores. This task is also called model failure prediction and highly depends on meaningful confidence estimates. First, the data drift robustness of these calibration methods combined with three different model architectures is evaluated. Two datasets from the friction stir welding domain containing realistic forms of data drift are introduced for this benchmark. Afterward, the methods' impact on model failure prediction performance is assessed. Findings confirm the positive influence of well-calibrated models on model failure prediction tasks, highlighting the need to look beyond classification accuracy during model selection. Moreover, transformer-based models and the WASAM technique were found to improve robustness to data drift, regarding the classification performance as well as obtaining useful confidence estimates.

Keywords Deep learning · Quality monitoring · Friction stir welding · Data drift · Calibration · Model failure prediction

Introduction

Due to their ability to extract meaningful features from high-dimensional data like images, deep neural networks (DNN) show high potential for the automation of optical quality monitoring tasks (Wang et al., 2018). Based on image captures of a specific region of interest, a product can be classified into *OK* or *nOK* (not OK) based on visible quality indicators. Examples include the friction stir welding process, where Hartl et al. (2019) showed that DNN, i.e., convolutional neural networks (CNN), are well suited to classify images of weld seams according to the welds' surface properties. Other applications include, e.g., the hairpin welding process in electric drive production (Mayr et al., 2022; Vater et al., 2021), optical inspection processes for remanufacturing scenarios (Nwankpa et al., 2021; Saiz et al., 2021), the electronics industry (Ebayyeh & Mousavi, 2020), or additive manufacturing processes (Vaghefi et al., 2024).

Although the mentioned works clearly show the potential of DNN for different quality monitoring applications, the investigations are usually conducted on fixed-size datasets, where training and testing data are independent and identically distributed. In practice, however, transferring a quality monitoring application to a different workstation or introducing new product variants can lead to influencing factors that change the optical appearance of the product and relevant defect patterns. In this case, the processed samples

✉ Johannes C. Bauer
johannes.bauer@iwb.tum.de

[1] Institute for Machine Tools and Industrial Management (iwb), Technical University of Munich, Boltzmannstraße 15, 85748 Garching, Germany

[2] Fraunhofer Institute for Casting, Composite and Processing Technology IGCV, Am Technologiezentrum 2, 86159 Augsburg, Germany

may no longer be distributed independently and identically to the training data. Such a shift in the data distribution is also referred to as data or concept drift and can significantly reduce the performance of a model (Lu et al., 2018; Widmer & Kubat, 1996). Especially in high-mix, low-volume production, such changes can happen quite frequently and also unknowingly, e.g, due to tool wear or changes in the environment. As a result, new models would have to be trained using new data on a regular basis to maintain a sufficient classification performance. This usually comes with significant efforts, e.g., for manual labeling, which currently hinders the application of DNN in such scenarios.

These efforts could be reduced by assessing the expected reliability of individual model predictions. This way, predictions with a high chance of being false could be filtered out for further inspection and subsequent adaptation of the model (Bauer & Daub, 2023). Technically, this could be realized by setting an acceptance threshold to the output class probability scores, also called confidence scores, of the model (Hendrycks & Gimpel, 2017). Such procedures are referred to as model failure prediction or misclassification detection. Unfortunately, these confidence scores are often not meaningful in practice and DNN tend to state high confidence scores for false predictions (Guo et al., 2017). So-called calibration approaches tackle this problem by aligning the confidence scores of the model with the actual prediction accuracy in the long run. Therefore, a well-calibrated model could help to avoid making false predictions and to increase the robustness of the overall quality monitoring system. This is in contrast to approaches like domain adaptation or continual learning, which aim to adapt the model itself to the data drift and rely on already collected representative datasets from the new data distribution.

Many different calibration methods have been proposed in scientific literature, but so far no common standard has emerged. Individual methods show different strengths and weaknesses regarding applicability, performance, or robustness and different issues remain. In fact, common calibration approaches are itself negatively affected by data drift (Ovadia et al., 2019). Moreover, the benefit of individual calibration methods for failure prediction has been questioned recently because these methods could not improve the separation of confidence values for correct and false predictions (Corbiere et al., 2019; Zhu et al., 2022). This stresses the need to use meaningful evaluation protocols targeted at the specific use case at hand since single evaluation metrics can only give incomplete information on a method's performance (Jaeger et al., 2023). The selection of suitable methods is further complicated by a lack of standardized benchmarking datasets that include data drift scenarios. In commonly used datasets like ImageNet-C (Hendrycks & Dietterich, 2019) the drift is generated synthetically and the dataset size is several orders of magnitude larger than they are typically in the manufacturing domain, where data is usually sparse and datasets small. So far, calibration and model failure prediction approaches have not been widely investigated for DNN-based quality monitoring applications – despite their potential to increase the robustness of such systems in practical use. As outlined above, it is still unclear how different calibration methods perform under data drift specific to quality monitoring applications and how they may help to perform effective model failure prediction.

To close this gap, this work aims to systematically evaluate the potential of calibration approaches for model failure prediction of DNN in quality monitoring applications. Its contributions can be separated into three areas. First, a comprehensive examination of methodological aspects regarding model failure prediction in quality monitoring settings is conducted and relevant parameters are identified. Second, two representative datasets from the friction stir welding domain are introduced that include real-world data drift, specific to quality monitoring applications. Third, an experimental investigation of the data drift robustness of different confidence calibration methods as well as DNN architectures is conducted, using the two datasets. Afterward, the impact of model calibration on model failure prediction performance is evaluated. The obtained results confirm the positive influence of suitable calibration methods on the model failure prediction performance. This helps to better identify and avoid potentially incorrect model predictions and opens up novel ways to improve the robustness of quality monitoring systems.

The remainder is structured as follows. In Sect. 2, fundamentals and related works are presented and discussed. Afterward, in Sect. 3, different sources of data drift are identified and the two evaluation datasets, displaying such drift, are characterized. In Sect. 4, we first present methodological aspects of model failure prediction for quality monitoring applications. Based on these explanations, the potential of calibration methods in this context is experimentally evaluated. Finally, a conclusion and outlook are given in Sect. 5.

Fundamentals and related works

Since friction stir welding serves as an exemplary use case for quality monitoring applications in this work, its fundamentals are briefly explained first before an overview of existing calibration methods is presented. Afterward, works on applications of calibration methods in the manufacturing domain are discussed.

Fundamentals of friction stir welding

Friction stir welding describes a solid-state joining process well-suited for aluminum alloys, which are difficult to join by

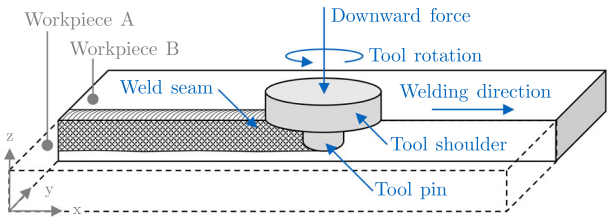


Fig. 1 Schematic visualization of the friction stir welding process, adapted from Mishra and Ma (2005); The rotating tool is moved along the joint trajectory and the workpieces are joined by mixing the materials in their plastic state

conventional welding processes (Thomas, 1998). By inserting a rotating tool in the joint area of two workpieces and moving it along the joint trajectory, frictional heat is created, and materials are mixed in their plastic state (Mishra & Ma, 2005). The process is visualized in Fig. 1.

One of the most influential aspects of process development is the tool design, e.g., the size of the tool shoulder and pin or the shoulder geometry, which plays a critical role in material flow (Mishra & Ma, 2005). Other relevant process parameters are, e.g., the tool rotation rate or the traverse speed along the trajectory (Mishra & Ma, 2005). Unsuitable process parameters or disturbances during the process can lead to different irregularities visible on the seam's surface (Zettler et al., 2010). This includes, e.g., irregular seam widths, axial cracks due to an insufficient connection between workpieces, or seam underfill (DIN EN ISO 25239-5, 12/2020; Hartl, 2021). Other visible defect types are excessive burr formation at a seam's edges, referred to as toe flash, or sheared-off particles on the seam's surface, also called surface galling (DIN EN ISO 25239-5, 12/2020; Hartl, 2021). Exemplary weld seam images are shown in Fig. 2.

Calibration of neural networks

For a classification task with n classes a DNN outputs a vector $\mathbf{v} \in \mathbb{R}^n$. Using the softmax function, the so-called softmax probabilities or confidence scores are obtained for each of the n classes. The prediction then corresponds to the class with the highest confidence score. A model can be called well-calibrated if the confidence scores of its predictions correspond to the actual prediction accuracy in the long run (Dawid, 1982). For example, if a calibrated model predicts that a set of inputs x belongs to class y with a confidence of 0.7, then we would expect that 70% of the inputs indeed belong to class y. As mentioned earlier, this is often not the case for DNN and models are overconfident in their predictions (Guo et al., 2017). Therefore, various methods for improving model calibration have been proposed in the last years. These methods can be divided into so-called post-hoc methods and intrinsic methods.

Post-hoc calibration refers to methods adjusting the confidence scores of an already existing model. The parameters for this adjustment process are optimized on an additional, held-out calibration dataset while the model parameters stay fixed. One of the simplest but most used approaches is temperature scaling (TS) (Guo et al., 2017). This method scales the model outputs by a single scalar parameter $T > 0$. Since scaling does not change the maximum of the softmax function, the predicted class remains unchanged and the method can be considered as accuracy preserving. Other well-known approaches are histogram binning (Zadrozny & Elkan, 2001) and isotonic regression (Zadrozny & Elkan, 2002), where samples are sorted into mutually exclusive bins based on their uncalibrated confidence scores. Afterward, calibrated scores are computed for each bin and assigned to all its samples. Some more recent and advanced methods include calibration using splines (Gupta et al., 2021), Dirichlet calibration (Kull et al., 2019), mutual information maximization-based binning (Patel et al., 2021), and intra order-preserving functions (Rahimi et al., 2020).

While common post-hoc methods based on scaling and binning-based approaches yield well-calibrated confidence scores in in-distribution scenarios, they can actually harm calibration when evaluated under data drift (Ovadia et al., 2019; Tomani et al., 2021). Although introducing perturbed images in the calibration set reduces this phenomenon, assumptions on potentially occurring drift scenarios would be required (Tomani et al., 2021). Tomani et al. (2023) tackle this problem by incorporating data density estimates into the scaling process. These density estimates are obtained by measuring the Euclidean distance of a sample x to its k^{th} nearest neighbor in the training data in the latent space of the model. However, scaling-based methods can be considered less relevant in terms of failure prediction or misclassification detection since these do not improve the separability of confidence scores for false and correct model predictions (Corbiere et al., 2019).

Intrinsic calibration methods are applied directly during model training. Gal and Ghahramani (2016) approximate Bayesian models by applying dropout before the weight layers of the model. Dropout randomly sets a specified fraction of weights to zero. Inference is then performed by averaging over multiple forward passes. This approach is commonly called Monte Carlo dropout or, if dropout is only inserted before the model's last classification layer, last-layer dropout (LLDo). Alternatively, ensemble models can be used (Lakshminarayanan et al., 2017). Such ensembles combine the predictions of multiple DNN trained individually for the same task.

Other intrinsic methods introduce regularization to the training process. For example, label smoothing (Szegedy et al., 2016) can improve calibration by manipulating the training labels, taking a small portion of the true class' prob-

ability and assigning it uniformly to the remaining classes (Mueller et al., 2019). Similarly, mixup (Zhang et al., 2020) improves calibration by randomly combining samples in the training data. Also, focal loss (Lin et al., 2017) can have a positive effect, as shown by Mukhoti et al. (2020). In Moon et al. (2020) a so-called correctness ranking loss (CRL) is introduced, specifically designed to regularize class probabilities to be better confidence estimates. Although the mentioned methods do reduce calibration errors, Zhu et al. (2022) recently questioned their contribution to model failure prediction. Instead, they highlight the positive influence of so-called flat-minima techniques like stochastic weight averaging (SWA) (Izmailov et al., 2018) or sharpness-aware minimization (SAM) (Foret et al., 2021). These approaches aim to find broader or flatter minima in the loss landscape during training. According to Zhu et al. (2022), this leads to a better separation between confidence scores of correct and false model predictions and reduces the negative influence of data drift on model calibration.

It's worth noting that the methods presented by Gal and Ghahramani (2016) and Lakshminarayanan et al. (2017), which are based on ensemble models or approximate Bayesian methods, also belong to the field of uncertainty estimation methods (Gawlikowski et al., 2023), forming some kind of intersection between uncertainty estimation methods and calibration methods. Uncertainty estimation techniques allow a differentiation between model uncertainty (e.g., due to overfitting of the training data) and data uncertainty (e.g., due to noise in the training data). By reducing the amount of model uncertainty included in the confidence scores, calibration is improved as well. In contrast, the other presented regularization-based calibration methods combine model and data uncertainty in the confidence score (Gawlikowski et al., 2023). For a more detailed explanation of uncertainty in DNN, its sources, and estimation methods readers may refer to Gawlikowski et al. (2023). However, uncertainty estimation methods also face different drawbacks, like longer training and inference times or the requirement of suitable priors, rendering them more complicated for practical use.

Application of calibration methods in the manufacturing context

Research on the utilization of calibration methods in manufacturing contexts, particularly for quality monitoring applications, is still relatively sparse. Nevertheless, the topic is taken up in some recent works, which will be discussed in the following.

Rožanec et al. (2023), propose a method for calibration assessment without ground truth data to improve active learning methods in a quality monitoring setting. The experiments show that the proposed label-free metrics capture relevant data otherwise summarized by common calibration metrics, which allowed a reduction of the labeling effort during the active learning process on an exemplary dataset. Cramer et al. (2022) use a Bayesian neural network to obtain uncertainty estimates for a predictive quality application. The model predicts the quality characteristics of injection molding parts based on process parameters. Although the estimated uncertainties are too high for actual industrial use, the overall method can be considered promising. Authors in Rathnakumar et al. (2023) incorporated a Bayesian neural network and dropout into a CNN-based model for crack segmentation. This way, detection performance is improved and uncertainty as well as the calibration error are reduced, compared to a fully convolutional baseline. Similarly, Pyle et al. (2022) investigate dropout and ensemble methods for uncertainty estimation applied to the problem of crack size prediction during inline pipe inspection. Models are trained on simulated data and evaluated on an experimentally generated test dataset. Ensemble models performed best, assigning high uncertainty to high error samples. In Kafunah et al. (2023), a method for combining the individual models' predictions in an ensemble is presented. The method generates a continuous probability distribution instead of outputting a point estimate. It is validated on two exemplary datasets and shows improved performance in terms of calibration and identification of synthetically altered samples.

Another line of work focuses on applications to monitor the state of manufacturing equipment. Similarly to previously mentioned approaches, the works presented by Lin and Li (2022), Maged and Xie (2022), Xiao et al. (2023), and Zhou et al. (2022) use Bayesian approximation techniques to obtain uncertainty estimates in order to improve remaining useful life prediction or machine failure detection.

As seen in Sect. 2.2, numerous options for calibrating DNN are currently being presented and discussed in scientific literature. While the above-mentioned works already underline the potential of calibration, i.e., uncertainty estimation methods, for different tasks in the manufacturing context, most of them focus on a single or a small subset of methods. Data drift scenarios are only marginally addressed so far, whereby the drift is usually generated synthetically. It remains unclear how different model architectures and calibration methods perform in quality monitoring settings, especially in the presence of real data drift, and how calibration performance translates to model failure prediction performance, e.g., in terms of avoided false predictions.

To address these gaps, two exemplary datasets are utilized that contain realistic and experimentally generated data drift scenarios for quality monitoring problems in friction stir welding. These form the basis for the subsequent evaluation of calibration methods and their potential for failure prediction and are presented in the next section.

Characterization of the datasets

Two datasets were utilized for the subsequent investigations. In contrast to popular benchmarks like ImageNet-C or CIFAR-C, the drift is not introduced synthetically by altering images of the validation set but generated from real-world experiments. To generate realistic shifts in the data distribution, possible influencing factors in quality monitoring were defined first using the example of friction stir welding processes.

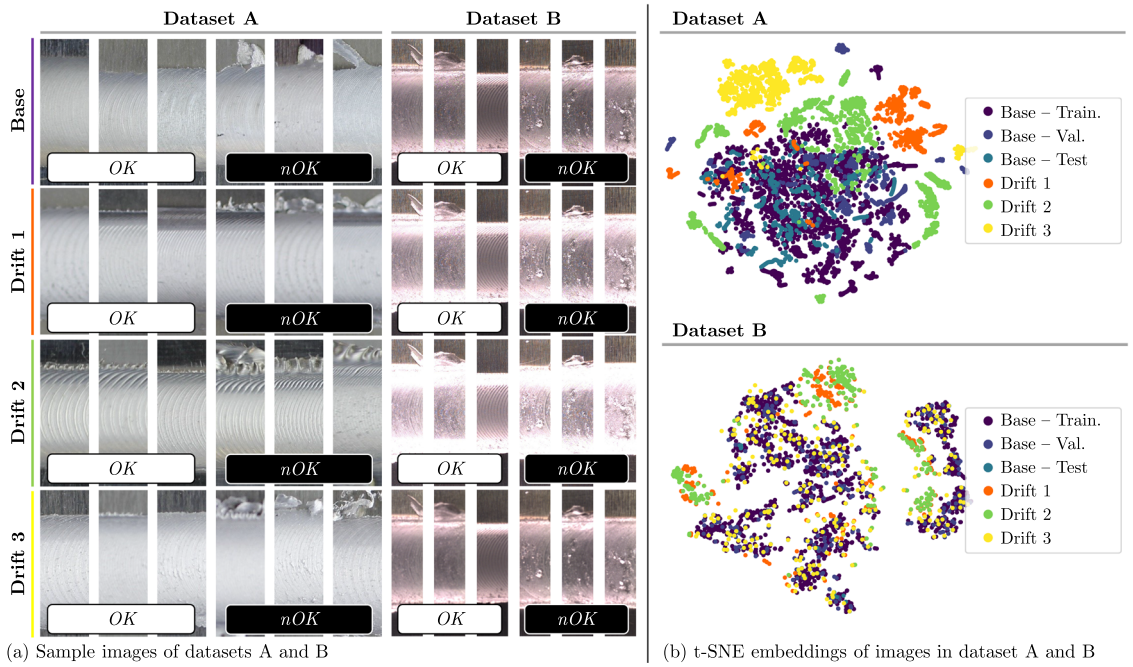
Influencing factors that may lead to data drift can be associated with three categories: (1) product and process-related factors, (2) monitoring system-related factors, and (3) post-processing-related factors.

(1) Product and process-related factors are deviations in the actual manufacturing process that change the optical appearance of *OK* and *nOK* samples. In the case of friction stir welding, this may include a change of material, different welding or traverse speeds, tool rotation rates, or tool geometries. Data drift due to these factors should be easy to avoid since such changes should not be introduced unknowingly. However, this does not apply to tool wear or process disturbances like an offset between workpieces. Furthermore, increasing model robustness against them is rather difficult since they cannot be replicated synthetically using common data augmentation techniques.
(2) Monitoring system-related factors are caused by the components of the sensory setup used for data acquisition. In optical quality monitoring, this may include the camera, the optics, but also the lighting conditions. Data drift can appear due to a substitution or degradation of the camera sensor, a transfer of the system to another machine, changes in the environment, or blurred images due to a failure or a faulty setup of the camera optics. This also includes disturbances due to dust or smoke, which take on a special role since they can also be caused by the process itself. On the one hand, the influence of such factors could be mitigated by carefully designed data augmentation during training. On the other hand, they may appear unknowingly, making it harder to avoid them completely.
(3) Post-processing-related factors refer to changes in the digital processing pipeline, e.g., changed image formats, rescaling operations, or normalization techniques, before the images are fed to the model. Although this seems unlikely, such errors may occur in practice, especially when scaling up such monitoring systems to a large number of machines or managing multiple monitoring systems in parallel. Nevertheless, they can be avoided by introducing suitable IT architectures, such as proposed by, e.g., Raffin et al. (2022). Hence, the two datasets focus on drift from categories (1) and (2).

Dataset A utilizes image data generated by Hartl et al. (2019) and contains drift due to product and process-related factors. The 112 welding samples in the dataset were originally manufactured in seven groups, varying the tool geometry, aluminum alloy, and material thickness. The images of a weld seam are processed in small sections of 100 to 300 pixels, resulting in a total of 858 to 1532 RGB images per manufactured group and 8460 images in the overall dataset. Therefore, we split the dataset according to these seven groups to generate data drift, choosing groups 1, 2, 6, and 7 as the base dataset. From this base dataset, 15% of weld seams are used as validation, and another 15% are used as in-distribution test data. Groups 3, 4, and 5 were used as drifted test datasets. The selection was based on the welding parameters, trying to provoke data drift of varying degrees, but otherwise did not follow any particular intention. For more detailed information on the dataset we refer to Hartl et al. (2019) as well as the Appendix. Images are classified according to the visibility of excessive burr formation, resulting in approximately the same number of *OK* and *nOK* samples. Exemplary images are presented in Fig. 2a.

Dataset B addresses data drift due to deviations in the monitoring system. Therefore, 103 weld seams are captured using an RGB camera. Similarly to dataset A, each weld seam is captured in multiple smaller sections, resulting in an overall amount of 1722 images. This process is repeated under varying influencing factors, yielding a total of n times 1722 images for n influencing factor combinations (with $n = 4$ in this case). To ensure an accurate positioning of the camera, it is mounted on a Franka Emika 6-axis robot arm that moves the camera along the welding trajectory. Lighting conditions are controlled via two LED panels mounted in parallel to the weld seam, which provide diffuse illumination. One basis and three drift scenarios are included in the dataset. The drift scenarios contain two levels of increased brightness as well as blurred images due to a distorted focus of the optics. The images are labeled according to the visibility of excessive surface irregularities with a roughly even distribution of *OK* and *nOK* labels. From the images of the base dataset, 15% are split as a validation set, and another 15% are used as in-distribution test data. The images are downsized to a resolution of 50 to 250 pixels. Exemplary images are shown in Fig. 2a.

To underline the visual difference between images in the respective base datasets and the drifted datasets, 2-dimensional t-SNE (t-distributed stochastic neighbor) embeddings are generated based on downsampled versions of the images and shown in Fig. 2b. t-SNE (van der Maaten & Hinton, 2008) is a popular technique for dimensionality reduction to visualize high-dimensional data. Although this



(a) Sample images of datasets A and B

(b) t-SNE embeddings of images in dataset A and B

Fig. 2 Exemplary images for both datasets are shown in (**a**), including the respective subsets for in-distribution (Base) and drifted data (Drift 1, 2, and 3). t-SNE embeddings of the samples in both datasets are visualized in (**b**). While drift in dataset A is caused by different process parameters, drift in dataset B comes from deviations in the monitoring setup (best viewed in color) (Color figure online)

should not be interpreted as a quantitative measure of the drift's severity, the deviation between drifted and non-drifted samples is clearly visible in the embedding space, except for the third drifted subset of dataset B. In the latter, the blur does not appear to be reflected in the created embeddings, given that the lighting conditions are unchanged from the base dataset. However, the blur's influence on model performance is noticeable, as seen in Table 3. Additional information on the data drift and its severity can be found in the Appendix.

Evaluating the potential of calibration methods for model failure prediction

To evaluate the potential of calibration methods for model failure prediction of quality monitoring applications it is first explained in more detail how model failure prediction can be implemented in manufacturing scenarios. Then, in Sect. 4.2, different model architectures and calibration methods are benchmarked on the two datasets presented in the previous section to evaluate their robustness to concept drift in the manufacturing domain and assess their suitability for model failure prediction. Afterward, in Sect. 4.3, their actual impact on model failure prediction performance is measured.

Methodological aspects of model failure prediction in quality monitoring

As mentioned earlier, a straightforward way to identify false model predictions is to set an acceptance threshold c_t on the confidence values of predictions and filter out all values below the threshold. Such filtered-out samples can then be manually inspected to obtain correct information on their quality. In the case of quality monitoring applications, model failure prediction may pursue different objectives. Depending on the considered use case, it can be likely that, e.g., misclassified *nOK* samples must be avoided at all costs to adhere to high-quality standards. In turn, it may be possible that the avoidance of unnecessary scrap is a priority and the amount of falsely classified *OK* samples should be reduced. Such scenarios may benefit from class-specific thresholds to adjust the rate of inspected samples. On the other hand, the rate of inspected samples is relevant itself, especially if the inspection is carried out manually. Depending on the cycle time, only relatively small inspection rates may be possible with fixed personnel capacities and efforts for manual inspection should be reduced whenever possible. The general workflow of model failure prediction, including relevant parameters and objectives, is visualized in Fig. 3.

Objectives that quantify how successful the model failure prediction is carried out include the resulting classification performance after inspection as measured by metrics like the

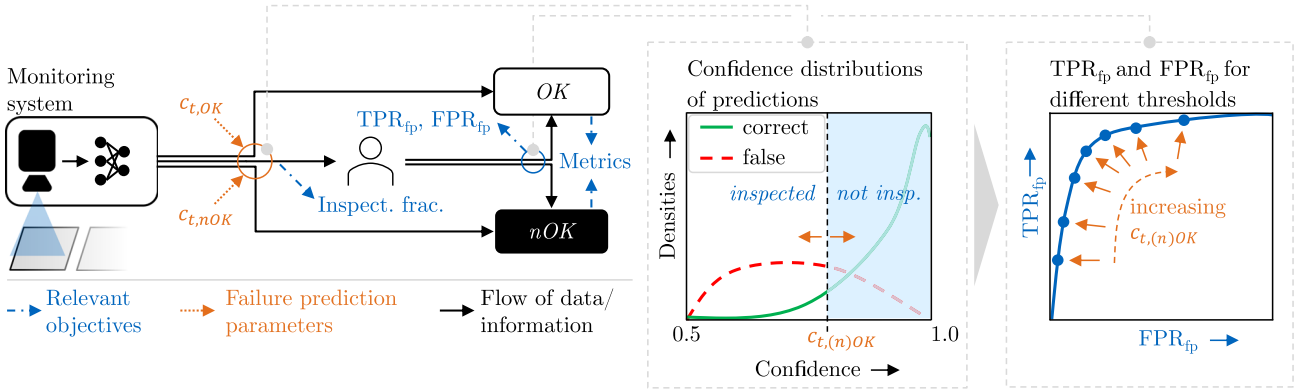


Fig. 3 Model failure prediction workflow; The threshold parameters $c_{t,OK}$ and $c_{t,nOK}$ are used to filter model predictions based on their confidence and adjust the trade-offs between relevant objectives. These trade-offs are visualized in the dashed boxes. Higher confidence thresholds $c_{t,(n)OK}$ lead to a greater share of identified false predictions (TPR$_{fp}$) but also increase the overall fraction of samples for additional inspection (Inspect. frac.) and the share of unnecessarily inspected corrected predictions (FPR$_{fp}$). These influence the overall system performance and resulting classification metrics

prediction accuracy, etc. or the overall fraction of samples to inspect. In the following, the latter will be also referred to as the *inspection fraction* and is defined by the number of all samples that have a confidence below c_t divided by the overall number of samples. Another objective is the fraction of actually identified false predictions in the total number of false model predictions, which should be as large as possible. In the following, this is denoted as the true positive rate for failure prediction TPR$_{fp}$. In contrast, the share of actually correct model predictions that are below the threshold (and therefore identified as false) will be referred to as the false positive rate for failure prediction FPR$_{fp}$. This rate should be as low as possible. For example, if a model makes a total of 100 false predictions on a given dataset and 60 of these false predictions have a confidence score below a defined confidence threshold c_t, the resulting TPR$_{fp}$ would be 0.6.

Hence, the trade-offs between these objectives can be adjusted by setting suitable confidence thresholds $c_{t,OK}$ and $c_{t,nOK}$ for the individual classes. These can be found empirically using a representative dataset and a given model. If a stream of incoming images is processed, the thresholds can be set dynamically as well, e.g., to reach a desired inspection fraction based on the confidence scores of the n last model predictions. In practice, multiple variations of this workflow would be possible, e.g., a scenario where all *nOK* samples are inspected by default and reworked if necessary, corresponding to a fixed $c_{t,nOK}$ of 1.0.

It may be noted that applying a threshold to the model's confidence scores is not the only option to identify potentially false model predictions. Alternative approaches could, e.g., measure the distance of samples in the embedding space, similar to the work of Tomani et al. (2023), or learn an additional model that predicts if a classification model's prediction is likely to be correct or not (Corbiere et al., 2019). However, these would increase the computation efforts during inference. We chose to focus on the confidence scores instead for the sake of simplicity and easier practical application.

Since the model failure prediction performance also depends on the performance of the underlying model, in the next section, multiple model architectures and calibration methods are benchmarked to identify promising approaches.

Benchmarking selected calibration methods and model architectures

In the following, we present results of a benchmark of selected model architectures and calibration methods on the two friction stir welding datasets to assess their robustness to data drift and relevant characteristics for failure prediction. It is first reported on the methodology and implementation before the results are presented and discussed.

Method and implementation

The experiments were performed using the Python programming language and implementations are based on the framework PyTorch (Paszke et al., 2019). We will first present the chosen model architectures and calibration methods. Afterward, the basic training and evaluation protocol, including relevant metrics, is described.

Model architectures: Three different model architectures are chosen for evaluation with the aim of including a diverse spectrum of architectures. These include the ResNet-18 architecture developed by He et al. (2016), which is a comparably small CNN but uses the popular ResNet approach that introduced residual connections. The EfficientNetV2-M (Tan & Le, 2021) architecture represents larger and more modern CNN architectures that show improved performance com-

pared to the ResNet family. Furthermore, SwinV2-B (Liu et al., 2022) is included as a state-of-the-art transformer-based DNN, which is not based on convolutional layers but uses the attention mechanism. It should be noted that the primary aim is not to compare the specific model architectures, but rather to include representatives of different architectural approaches. All models are initialized with pre-trained weights from the ImageNet-1k dataset as provided by PyTorch's torchvision package.

Calibration methods: The evaluated calibration methods include the intrinsic methods LLDo (Gal & Ghahramani, 2016), CRL (Moon et al., 2020), and the flat-minima technique WASAM (weight-averaged sharpness-aware minimization) (Kaddour et al., 2022), which combines aspects of SWA and SAM. We focus on intrinsic calibration techniques due to their higher potential for model failure prediction and easier practical application compared to post-hoc methods and uncertainty estimation techniques (c.f., Sect. 2.2).

In the following, the methods are explained in more detail. Further information on the used hyperparameters can be found in the Appendix.

LLDo represents a well-known calibration approach and is also included because of its conceptual simplicity. It is implemented by adding a dropout layer before the final classification layer of the network. During inference, the dropout layer randomly sets a fraction $p \in [0, 1]$ of weights to zero. By averaging the outputs of n such stochastic forward passes a calibrated output vector is obtained.

CRL is chosen since it is a state-of-the-art calibration approach. It adds an additional regularization term to the loss function. It aims to improve the confidence scores in terms of ordinal ranking of predictions according to confidence. This renders the method well-suited for model failure prediction tasks. The additional loss term $\mathcal{L}_{CRL}$ is weighted by a hyperparameter λ and calculated as follows. For a set of training sample pairs $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{D}$ an ordinal ranking relationship $\mathcal{L}_{CRL}(\mathbf{x}_i, \mathbf{x}_j)$ is calculated as stated in Eqs. (1) and (2) (Moon et al., 2020),

$$\mathcal{L}_{CRL}(\mathbf{x}_i, \mathbf{x}_j) = \max(0, \phi) \tag{1}$$

$$\phi = -g(r_i, r_j)(\text{conf}_i - \text{conf}_j) + |r_i - r_j| \tag{2}$$

where r_i is the fraction of correct prediction events of $\mathbf{x}_i$ over the total number of examinations during the previous training process, conf_i is the confidence score for $\mathbf{x}_i$, and $g(r_i, r_j)$ is 1 if $r_i > r_j$, 0 if $r_i = r_j$, and -1 otherwise.

The use of WASAM is inspired by the findings of Zhu et al. (2022). The method performs SWA on the parameters obtained by SAM. The SWA process is started after a defined number of training epochs has passed. It then computes a cumulative moving average of the model weights as stated in Eq. (3) (Izmailov et al., 2018),

$$\boldsymbol{\theta}_t^{SWA} = \frac{\boldsymbol{\theta}_{t-1}^{SWA} \cdot l + \boldsymbol{\theta}_t}{l + 1} \tag{3}$$

where l counts the number of previous SWA updates, $\boldsymbol{\theta}_t$ represents the weights of the current epoch, and $\boldsymbol{\theta}^{SWA}$ are the averaged weights. SAM, on the other hand, guides the optimization of the model parameters $\boldsymbol{\theta}$. It first finds a worst-case parameter perturbation $\boldsymbol{\epsilon}$, with $||\boldsymbol{\epsilon}||_2 \leq \rho$, that maximizes the loss value. The value of ρ is set as a hyperparameter. The method then minimizes the loss with regard to the perturbed weights $\boldsymbol{\theta} + \boldsymbol{\epsilon}$. It therefore optimizes the model parameters by solving the minimax problem stated in Eq. (4) (Foret et al., 2021),

$$\min_{\boldsymbol{\theta}} \max_{||\boldsymbol{\epsilon}||_2 \leq \rho} \mathcal{L}(\boldsymbol{\theta} + \boldsymbol{\epsilon}) \tag{4}$$

All three methods were reimplemented based on available information and code published by the original authors. Since CRL and WASAM are based on regularization they do not increase the inference time compared to the baseline models, which is an advantage for practical application. Merely LLDo slightly increased the inference time from 0.008 to 0.014 seconds per sample (measured using a ResNet-18 and a single input image on a setup with a 24 core AMD Epyic CPU, 192 GB RAM, and a Nvidia RTX3090 GPU). The observed differences in training time are limited to a few minutes per run and should not be significant in practice.

Training and model selection: Models are trained using early stopping on the hold-out validation data, keeping the weights of the best-performing epoch. During training, the addition of random Gaussian noise and random vertical flips are performed for on-the-fly data augmentation when loading the images. The learning rate, as well as other method-specific hyperparameters, are selected on the validation set. More information on the hyperparameter selection can be found in the Appendix. The model-method combinations with the minimum validation set loss are then evaluated on the in-distribution test sets and the drifted datasets.

Metrics: Various metrics are used to evaluate the models, taking the methodological aspects of model failure prediction into account. Since both datasets A and B have an approximately balanced label distribution, the models' classification performance is primarily assessed using the classification accuracy and the area under receiver operating characteristic (AUROC) (Hanley & McNeil, 1982). Both represent commonly used metrics.

Calibration is assessed using the expected calibration error (ECE) (Naeini et al., 2015) and the Brier score (Brier, 1950), which are popular metrics for calibration assessment. The ECE measures how well the confidence scores match the prediction accuracy. Therefore, the predictions are sorted into

M equally sized bins according to their confidence scores and a weighted average over the absolute difference between accuracy and confidence is computed as stated in Eq. (5),

$$ECE = \sum_{m=1}^{M} \frac{N_m}{N} |\text{Acc}_m - \text{Conf}_m| \tag{5}$$

where Acc_m and Conf_m represent the accuracy and confidence for samples in bin m. N and N_m represent the total number of samples and the number of samples in bin m, respectively. The number of bins M is set to 15 following common practice. The Brier score measures how well a predicted probability vector $\mathbf{s}$ explains an observation $\mathbf{y}$. For K classes, the Brier score for an individual sample n can be calculated as stated in Eq. (6),

$$Brier_n = \sum_{k=1}^{K} (s_k - y_k)^2 \tag{6}$$

where s_k is the estimated probability for class k and y_k is either 0 or 1, depending on the true class of the sample.

Since successful model failure prediction relies on a clear separation of confidence scores for correct and false predictions (Zhu et al., 2022), we introduce a fourth metric to explicitly measure this separation, which is inspired by the AUROC calculation. As mentioned above, each defined confidence threshold $c_{t,(n)OK}$ comes with a rate of correctly identified false predictions TPR_{fp} and a rate of actually correct predictions that are filtered out as well FPR_{fp}, which are calculated as stated in Eqs. (7) and (8),

$$\text{TPR}_{fp} = \text{TP}_{fp}/\text{P}_{fp} \tag{7}$$

$$\text{FPR}_{fp} = \text{FP}_{fp}/\text{N}_{fp} \tag{8}$$

where TP_{fp} is the number of false predictions with a confidence below $c_{t,(n)OK}$, FP_{fp} is the number of correct predictions with a confidence below $c_{t,(n)OK}$, P_{fp} is the number of all false model predictions and N_{fp} is the number of all correct model predictions. By measuring the area that is defined by visualizing the TPR_{fp} over the FPR_{fp} for different thresholds we can quantify the separability of the confidence distributions of false and correct model predictions (c.f., Fig. 3). The separation score approaches 0.5 in case the confidence distributions of false and correct predictions completely overlap and 1.0 if the distributions can be completely separated.

Presentation and discussion of results

In the following, results regarding the different model architectures are presented and discussed first. Afterward, the different calibration methods are examined.

Model architectures: Table 1 shows the performance of the three model architectures ResNet-18, EfficientNetV2-M, and SwinV2-B on the in-distribution test data as well as the drift data. The scores for individual metrics represent an average of all calibration methods combined with the particular model architecture, to also take the robustness to different training regimes into account.

When only considering the performance on the in-distribution test data, results are relatively mixed and no architecture can be considered as clearly superior. Notably, the relatively small ResNet-18 with less than 12 Mio. parameters performs more or less on par with the considerably larger EfficientNet and Swin transformer with about 54 and 88 Mio. parameters, respectively. However, on the drift data, performance drops significantly for all considered metrics and the differences between model architectures increase. The larger architectures now show consistently better performance regarding classification as well as calibration and separation. The Swin transformer architecture clearly shows the best robustness to data drift on both datasets.

Results are in line with findings of Hendrycks and Dietterich (2019) showing that larger and deeper CNN architectures are more robust to image perturbations. Similarly, Minderer et al. (2021) also point out that the decay of calibration with data drift is less pronounced for non-convolutional architectures like transformers as it is in CNN. Although, our results do not confirm such differences regarding calibration for the in-distribution test data. According to Minderer et al. (2021), these differences between transformer-based models and CNN can not be fully explained by model size and pre-training but may be attributed to the differences in model architecture. Overall, the results stress the relevance of larger and transformer-based model architectures, even when smaller networks seem to keep up on in-distribution validation or test data for quality monitoring tasks.

Calibration methods: Table 2 compares the performance of the selected calibration methods on the two datasets. Results are averaged over different model architectures to favor methods that are robust across different types of DNN. Again, the results on the drift datasets are aggregated for easier comparison.

As seen before, the difference between methods on the in-distribution test data is rather low and no method can be considered as a clear favorite. Nevertheless, LLDo and WASAM show relatively consistent results regarding classification performance and calibration, especially on dataset B. While all methods reduce the ECE and Brier score and increase the separation on the in-distribution test data of dataset B, ECE and Brier scores of the baseline models are actually among the lowest for dataset A. Shifting the focus to the drift data, a clear decrease in classification performance as well as calibration and separation can be noticed for all methods. However, now WASAM shows the best per-

Table 1 Average performance of different model architectures on datasets A and B for all methods, quantified by the metrics classification accuracy (acc.), AUROC, ECE, Brier, and confidence separation (sep.). The best-performing models per dataset and metric are marked bold. Results for the three drifted subsets of each dataset (c.f., Fig. 2) are averaged for easier comparison. Selected performance metrics for individual subsets are stated in the Appendix

Dataset	Models	In-distribution test data					Drift data (averaged)				
		Acc.↑	ECE↓	AUROC↑	Brier↓	Sep.↑	Acc.↑	ECE↓	AUROC↑	Brier↓	Sep.↑
A	ResNet-18	**0.954**	0.033	0.983	0.084	0.800	0.767	0.117	0.860	0.339	0.757
	EffNet-M	0.944	0.028	**0.989**	0.089	**0.890**	0.813	**0.066**	0.885	0.270	0.779
	Swin-B	**0.954**	**0.020**	0.982	**0.074**	0.822	**0.827**	0.075	**0.918**	**0.252**	**0.816**
B	ResNet-18	0.931	**0.039**	0.983	**0.103**	**0.911**	0.853	0.085	0.944	0.231	0.793
	EffNet-M	0.911	0.053	0.968	0.142	0.859	0.878	**0.066**	0.948	0.185	0.833
	Swin-B	**0.933**	0.046	**0.984**	0.108	0.880	**0.906**	**0.066**	**0.967**	**0.152**	**0.848**

Table 2 Average performance of different calibration methods on the two datasets A and B across all model architectures, quantified by the metrics classification accuracy (acc.), AUROC, ECE, Brier, and confidence separation (sep.). The best method per dataset and metric is marked bold. Results for the three drifted subsets of each dataset (c.f., Fig. 2) are averaged for easier comparison

Dataset	Method	In-distribution test data					Drift data (averaged)				
		Acc.↑	ECE↓	AUROC↑	Brier↓	Sep.↑	Acc.↑	ECE↓	AUROC↑	Brier↓	Sep.↑
A	Baseline	0.952	**0.017**	0.984	0.079	0.827	0.802	0.085	0.878	0.287	0.780
	LLDo	0.943	0.028	**0.988**	0.090	**0.872**	0.760	0.154	0.887	0.375	0.770
	CRL	**0.954**	0.033	0.983	0.082	0.820	0.819	0.071	**0.903**	0.262	0.792
	WASAM	0.952	0.021	0.986	**0.077**	0.856	**0.830**	**0.057**	0.902	**0.241**	**0.804**
B	Baseline	0.915	0.054	0.971	0.141	0.849	0.879	0.077	0.948	0.194	0.804
	LLDo	**0.940**	**0.034**	**0.987**	0.094	0.896	0.871	0.078	0.954	0.200	0.828
	CRL	0.932	0.044	0.982	0.103	0.899	0.864	0.094	0.952	0.214	0.835
	WASAM	0.933	0.036	**0.987**	**0.091**	**0.936**	**0.903**	**0.043**	**0.968**	**0.146**	**0.864**

formance among the methods across almost all metrics on both datasets.

The results reaffirm the positive influence of flat-minima techniques like WASAM, also stated by Zhu et al. (2022), for the deployed quality monitoring-specific friction stir welding datasets. LLDo, on the other hand, drops significantly in performance in case of data drift, especially on dataset A. This matches the observation of Ovadia et al. (2019) that better accuracy and calibration on in-distribution data does not automatically lead to better performance under data drift. A possible explanation for the lower impact of data drift on the WASAM-trained models might be that WASAM (and the methods it combines) were originally designed to increase models' generalization robustness (Kaddour et al., 2022). This seems to be effective regarding the models' classification performance as well as their confidence calibration and separation when dealing with data drift. While CRL also regularizes the training process, improving a model's generalization robustness is not its primary objective. Although Moon et al. (2020) show that CRL can effectively improve the detection of out-of-distribution samples based on models' confidence scores, it seems to be less effective regarding the data drift included in our datasets. After all, the evaluated out-of-distribution detection tasks in Moon et al. (2020) deal with different kinds of distributional shift, compared to the datasets used in this study.

Main findings: The conclusion we draw from this benchmark is twofold:

(1) On the one hand, results stress the importance of a proper evaluation protocol before models are deployed in production. This means choosing relevant metrics beyond the classification accuracy to obtain a holistic performance estimate. Especially model failure prediction benefits from a good separation between confidence scores of false and correct predictions, which we show in more detail in Sect. 4.3. However, as can be seen in Fig. 4, this separation correlates significantly less with the optimized loss value than the classification accuracy. This leads to models that have high predictive performance but are less suited for failure prediction if separation is not explicitly measured. Besides suitable metrics, the selection of evaluation data is crucial. Ideally, some form of data drift should be included in the test dataset to estimate a model's robustness to such phenomena during deployment.

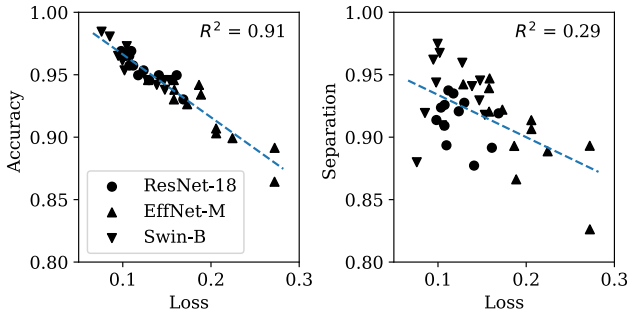


Fig. 4 Correlation of the validation loss with the classification accuracy and confidence separation on the validation set of dataset B; Indicators represent different model architectures trained without additional calibration techniques but varying learning rates. The dashed line visualizes a linear fit of the data points

(2) On the other hand, if an evaluation of the models' performance on drifted data is not possible, opting for transformer-based architectures and flat-minima techniques like WASAM will likely yield more robust models. Both methods showed improved robustness to data drift on both datasets, regarding classification performance as well as calibration and confidence separation metrics.

Impact on model failure prediction performance

In the following, we investigate how a model's benchmark performance is actually reflected in its model failure prediction effectiveness. In the most general case, the goal of model failure prediction can be formulated as minimizing the combined cost resulting from a certain rate of correctly identified (and therefore avoided) false model predictions TPR_{fp} and the corresponding effort for inspection represented by the overall *inspection fraction*. Therefore, we will focus on these objectives in the following. Based on the results of the previous section, mainly the Swin transformer models trained by the WASAM technique are evaluated.

Figure 5a visualizes the TPR_{fp} and the corresponding *inspection fraction* for selected models on the in-distribution test datasets of both datasets A and B. Confidence thresholds for both classes are set to the same values to treat false predictions of *nOK* and *OK* samples equally. Corresponding confidence distributions are shown in Fig. 5b.

It can be seen that for models trained with the WASAM technique, the rate of identified false model predictions is considerably higher at a given fraction of overall samples to inspect. For example, on dataset A with a threshold of $c_{t,(n)OK} = 0.8$, already 66.7% of false predictions could be avoided with an inspection fraction of 10.7%. This would increase the overall system accuracy to 98.2%, assuming a flawless manual inspection process. While the baseline models show clear peaks for high confidence values for false predictions (see Fig. 5b), these peaks are less pronounced or not visible at all for the WASAM-trained models. It should be noted that the models show similar performance metrics on the respective datasets, except for the separation metric. The exact values are stated in the Appendix. The presented results further underline the importance of well-separated confidence distributions and their assessment during model selection.

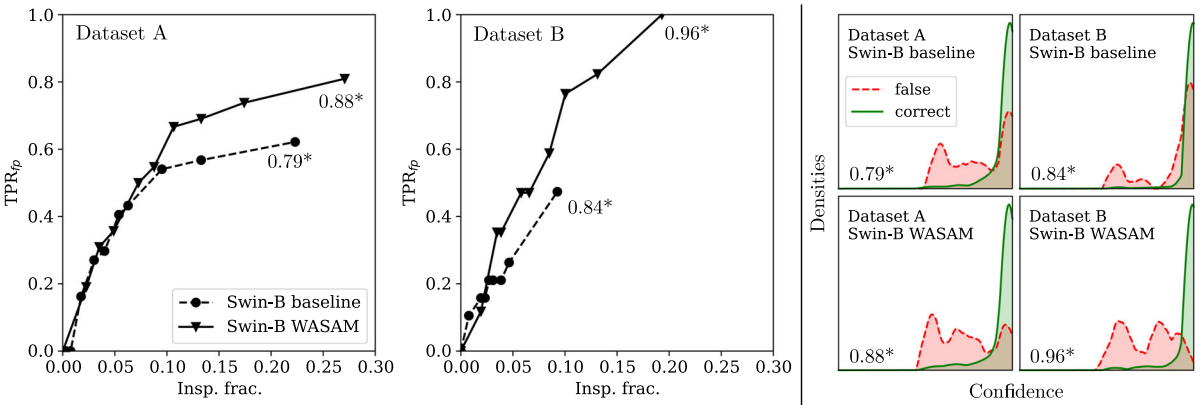
To evaluate the failure prediction performance in the presence of data drift, the TPR_{fp}, the *inspection fraction*, and the resulting overall *system accuracy* are examined for varying confidence thresholds on the individual subsets of dataset A. Results are visualized in Fig. 6.

With the previously chosen threshold of $c_{t,(n)OK} = 0.8$, up to 72.6% of false predictions would still be identified on the drift data (see Drift 1, 2, and 3 in Fig. 6). On the downside, the fraction of samples to inspect would rise up to 30.5%, along with the share of unnecessarily inspected correct predictions. However, the pure model accuracy (corresponding to $c_{t,(n)OK} = 0.5$) on the second drift scenario is already down to 82.2% (see also Table 3 in the Appendix), stating an urgent need for model adaptation.

The presented results clearly point out the positive influence of well-separated confidence distributions for failure prediction, as also mentioned in Zhu et al. (2022). One opportunity to assess if a model's confidence scores are well-separated is the separation metric presented in Sect. 4.2. If they are, a significant part of false predictions can be avoided. Comprehensibly, the effort for manual inspection rises noticeably as soon as the classification performance decreases with data drift. This poses a challenge for the practical application of the method since high manual inspection rates introduce additional costs. In turn, such an increase in low-confidence predictions could also serve as an indicator for reduced model performance due to data drift. Subsequently, measures to improve model performance again can be taken. Nevertheless, further studies would be necessary to provide reliable evidence and to also take existing data drift monitoring approaches into account. Another practical challenge is the definition of a suitable confidence threshold. As already mentioned in Sect. 4.1, choosing a dynamic threshold based on the confidence scores of former predictions, with the aim of maintaining a roughly constant rate of manually inspected samples, may be a promising option.

Conclusion and outlook

The presented study aims to evaluate the potential of calibration methods for model failure prediction in deep learning-based optical quality monitoring applications, including scenarios affected by data drift. To reach this goal, two datasets



(a) Avoided false model predictions vs. inspection efforts (*corresponding separation scores)

(b) Confidence distributions (*corresponding separation scores)

Fig. 5 Failure prediction performance of different methods on the in-distribution test sets (**a**) and corresponding confidence distributions (**b**); Indicators in (**a**) represent different confidence thresholds $c_{t,(n)OK} \in \{0.50, 0.55, \ldots, 0.95\}$ causing the respective values for TPR$_{fp}$ and inspection fractions. Confidence distributions of correct and false model predictions in (**b**) correspond to density histograms, smoothed by kernel density estimation for better visualization

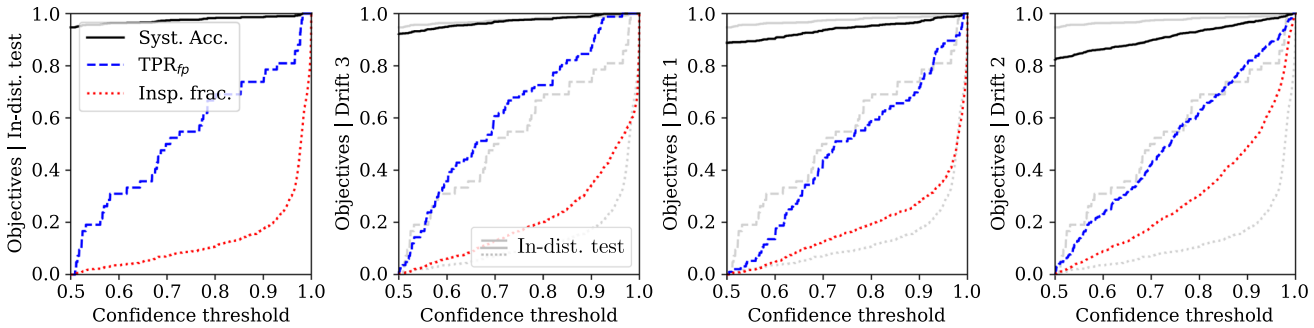


Fig. 6 Model failure prediction performance under data drift, quantified by the TPR$_{fp}$, inspection fraction, and resulting system accuracy, exemplarily shown by a Swin transformer trained with WASAM on dataset A; The grey lines resemble the in-distribution performance for easier comparison. Plots are ordered according to the model's classification performance on the respective subsets, decreasing from left to right

from the friction stir welding domain were generated. These include realistic forms of data drift that apply specifically to quality monitoring applications. After methodological aspects of model failure prediction in such settings were examined, different models and calibration approaches were systematically benchmarked on the two datasets, to assess their potential for model failure prediction.

The obtained results confirm the positive influence confidence calibration methods can have on model failure prediction. Furthermore, the benchmark and discussed methodological aspects may serve as a starting point for future research and the integration of such model failure prediction approaches into real-world quality monitoring applications.

In particular, results show that the transformer-based model architecture SwinV2-B and the flat-minima technique WASAM are more robust to data drift than other evaluated models and calibration approaches. Apart from an improved classification performance as well as improved ECE and Brier scores, they improve the separation of confidence scores for false and correct predictions on both datasets. This renders them well-suited for model failure prediction since the amount of correctly detected false model predictions for a given confidence threshold relies largely on a good separation of confidence scores. Thereby, up to 76% of false predictions could be avoided on the in-distribution test data of the two datasets, with about 10% of processed samples being subject to additional inspection. Such well-calibrated models improve the performance of the entire monitoring system and help to use human input in a more targeted way.

Although we were able to utilize two datasets that show a wide variety of representative data drift scenarios, both datasets stem from the friction stir welding domain. Since the presented methods and approaches are applicable to other quality monitoring scenarios as well future work could

address different manufacturing domains and sensor modalities to further validate the findings. The same applies to the evaluated model architectures. Although we have attempted to make a representative selection, further research should extend the investigation to other model architectures as well. Furthermore, the number of inspected samples rises significantly when model performance decreases in case of data drift, which can be challenging in practical applications. On the one hand, this underlines the importance of adapting the model itself to the changing data distribution. Active learning, domain adaptation, and continual learning approaches could be suitable for ensuring that the adaptation process is as effective as possible. On the other hand, monitoring the data distribution itself for signs of data drift can further contribute to increasing the reliability of deep learning-based quality monitoring applications in the industry.

Appendix

Dataset and data drift characterization

In Table 3, we state additional details on the data drift sources for dataset A and B and also list the number of *OK* and *nOK* samples per subset. In dataset A the drift is mainly induced by the different aluminium alloys and shoulder geometries of the weld tools used for the welding experiments. In dataset B the drift is created by varying the illumination intensity of the two 50 W LED light panels and disturbing the optics' focus.

Furthermore, the drift severity is quantified by reporting the classification accuracy and AUROC of two models on the drifted subsets. Additionally ECE and confidence separation are reported. The models were trained on the in-distribution training data. The ResNet-18 was trained without further calibration and the Swin transformer was trained using the WASAM method.

Hyperparameter selection

In the following, we state further details on the hyperparameter selection process regarding Sect. 4.2. Hyperparameters were chosen based on the validation set loss on the respective datasets. The loss function defaults to the cross-entropy loss implemented in PyTorch unless methods use a custom loss function, e.g., CRL. For further details on the methods we refer to the cited papers. The tuned as well as relevant fixed hyperparameters for each method-model combination are listed in Table 4. After preliminary tests were conducted to see if the method and model converged, a grid search over all hyperparameter combinations was performed, unless for WASAM were 24 random combinations were chosen due to computational limitations. The batch size for all training runs was set to 128. Images were normalized using the ImageNet mean $(0.485, 0.456, 0.406)$ and standard deviation $(0.229, 0.224, 0.225)$.

Additional performance metrics

Table 5 gives information on the performance of models used for failure prediction in Fig. 5.

Table 3 Overview of drift inducing parameters per subset for dataset A and B including the number of OK and nOK samples. Additionally we state the resulting accuracy (acc.), AUROC, ECE, and separation score (sep.) for a ResNet-18 without further calibration as well as a Swin transformer with WASAM trained on the in-distribution training data

Ds.	Subset	Parameters	Nr.	ResNet-18				Swin w/ WASAM			
			OK/nOK	Acc.	AUROC	ECE	Sep.	Acc.	AUROC	ECE	Sep.
A	In-dist test	EN AW-5754 H111/concave EN AW-5754 H111/spiral EN AW-6082 T6/concentric rings	426/432	0.959	0.983	0.020	0.788	0.947	0.984	0.020	0.880
	Drift 1	EN AW-6082 T6/concave	210/648	0.716	0.849	0.157	0.799	0.888	0.938	0.035	0.836
	Drift 2	EN AW-5754 H111/concentric rings	674/858	0.683	0.722	0.155	0.594	0.822	0.898	0.032	0.755
	Drift 3	EN AW-6082 T6/concentric rings	260/806	0.905	0.965	0.037	0.888	0.921	0.974	0.027	0.890
B	In-dist test	50% light intensity/no blur	158/101	0.923	0.977	0.035	0.886	0.934	0.990	0.025	0.957
	Drift 1	75% light intensity/no blur	158/101	0.907	0.965	0.057	0.840	0.942	0.986	0.031	0.904
	Drift 2	100% light intensity/no blur	158/101	0.819	0.908	0.065	0.735	0.923	0.974	0.033	0.861
	Drift 3	50% light intensity/blur	158/101	0.830	0.934	0.123	0.738	0.888	0.969	0.044	0.907

Table 4 Overview of tuned hyperparameters for individual methods. The stated search space was defined based on preliminary experiments for each method and model

Method	Hyperparameter	Value (s)
Baseline	Nr. epochs	20
	Learning rate (LR)	{1e−6, 5e−6, …, 1e−4}
	Scheduler	{None, Cosine}
	Optimizer	{Adam, AdamW}
LLDo	Dropout ratio p	0.5
	Nr. stoc. forwards n	100
	Nr. epochs	20
	LR	{5e−6, 1e−5, …, 5e−4}
	Scheduler	{None, Cosine}
	Optimizer	{Adam, AdamW}
CRL	Nr. epochs	20
	LR	{1e−6, 5e−6, …, 1e−4}
	Scheduler	Cosine
	Optimizer	{Adam, AdamW}
	λ_{CRL}	{0.05, 0.1, 0.25}
WASAM	Nr. epochs	{30, 40}
	Base optimizer	{Adam, SGD}
	Scheduler	Cosine
	LR_{base}	{1e−5, 5e−5, …1e−2}
	LR_{WASAM}	{1e−3, 1e−2, 1e−1}
	Momentum_{WASAM}	{0.9, 0.95, 0.99}
	ρ_{WASAM}	{1e−2, 5e−2, 1e−1}
	SWA start	{25%, 50%, 75%}

Table 5 Results of SwinV2-B trained with different methods on the in-distribution test data of both datasets. The corresponding failure prediction performance of the models is shown in Fig. 5. The best performance per method and metric is marked bold

Ds.	Method	Acc	ECE	AUROC	Brier
A	Baseline	**0.954**	**0.018**	0.979	**0.078**
	WASAM	0.947	0.020	**0.984**	0.081
B	Baseline	0.927	0.058	0.980	0.125
	WASAM	**0.934**	**0.025**	**0.990**	**0.085**

Acknowledgements JCB thanks Dr.-Ing. Roman Hartl for providing access to data used for the experiments.

Author Contributions JCB: Conceptualization, Methodology, Software, Investigation, Data Curation, Writing—Original Draft, Writing—Review & Editing, Visualization, Project administration. ST: Methodology, Writing—Review & Editing. FV: Methodology, Writing - Review & Editing. RD: Resources, Supervision, Funding Acquisition, Writing—Review.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data Availability The code as well as the data supporting the results of this study are available from the corresponding author upon reasonable request.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

Bauer, J. C., & Daub, R. (2023). Continuous adaptation of deep learning models for optical quality monitoring tasks. In *28th international conference on emerging technologies and factory automation (ETFA)* (pp. 1–4). IEEE. https://doi.org/10.1109/ETFA54631.2023.10275699

Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Review, 78*(1), 1–3. https://doi.org/10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2

Corbiere, C., Thome, N., Bar-Hen, A., Cord, M., & Perez, P. (2019). Addressing failure prediction by learning model confidence. *Advances in Neural Information Processing Systems, 32*, 2902–2913. arXiv:1910.04851.

Cramer, S., Huber, M., & Schmitt, R. H. (2022). Uncertainty quantification based on bayesian neural networks for predictive quality. In A. Steland and K-L. Tsui (Eds.), *Artificial intelligence, big data and data science in statistics* (pp. 253–268). Springer. https://doi.org/10.1007/978-3-031-07155-3_10

Dawid, A. P. (1982). The well-calibrated Bayesian. *Journal of the American Statistical Association, 77*(379), 605–610. https://doi.org/10.1080/01621459.1982.10477856

DIN EN ISO 25239-5. (2020). *Friction stir welding—Aluminium—Part 5: Quality and inspection requirements* (No. 25239-5:2020-12).

Ebayyeh, A. A. R. M. A., & Mousavi, A. (2020). A review and analysis of automatic optical inspection and quality monitoring methods in

electronics industry. *IEEE Access, 8*, 183192–183271. https://doi.org/10.1109/ACCESS.2020.3029127

Foret, P., Kleiner, A., Mobahi, H., & Neyshabur, B. (2021). Sharpness-aware minimization for efficiently improving generalization. In *9th international conference on learning representations (ICLR)* (pp. 1–20). arXiv:2010.01412.

Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *33rd international conference on machine learning (ICML)* (pp. 1050–1059). PMLR. https://doi.org/10.48550/arXiv.1506.02142

Gawlikowski, J., Tassi, C. R. N., Ali, M., Lee, J., Humt, M., Feng, J., & Zhu, X. X. (2023). A survey of uncertainty in deep neural networks. *Artificial Intelligence Review, 56*, 1513–1589. https://doi.org/10.1007/s10462-023-10562-9

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K.Q. (2017). On calibration of modern neural networks. In *34th international conference on machine learning (ICML)* (pp. 1321–1330). PMLR. arXiv:1706.04599.

Gupta, K., Rahimi, A., Ajanthan, T., Mensink, T., Sminchisescu, C., & Hartley, R. (2021). Calibration of neural networks using splines. In *9th international conference on learning representations (ICLR)* (pp. 1–19). arXiv:2006.12800.

Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology, 143*, 29–36. https://doi.org/10.1148/radiology.143.1.7063747

Hartl, R. (2021). *Monitoring and optimizing the surface quality of friction stir welds using machine learning: Dissertation*. utzverlag.

Hartl, R., Landgraf, J., Spahl, J., Bachmann, A., & Zaeh, M. F. (2019). Automated visual inspection of friction stir welds: A deep learning approach. *Proceedings of Spie 11059, Multimodal Sensing: Technologies and Applications, 11059*, 52–75. https://doi.org/10.1117/12.2525947

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Ieee conference on computer vision and pattern recognition (CVPR)* (pp. 770–778). IEEE. https://doi.org/10.1109/cvpr.2016.90

Hendrycks, D., & Dietterich, T. (2019). Benchmarking neural network robustness to common corruptions and perturbations. In *7th international conference on learning representations (ICLR)* (pp. 1–16). arXiv:1903.12261.

Hendrycks, D., & Gimpel, K. (2017). A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *5th international conference on learning representations (ICLR)* (pp. 1–12). arXiv:1610.02136.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., & Wilson, A.G. (2018). Averaging weights leads to wider optima and better generalization. In *34th conference on uncertainty in artificial intelligence (UAI)* (pp. 876–885). AUAI. arXiv:1803.05407.

Jaeger, P. F., Lüth, C. T., Klein, L., & Bungert, T. J. (2023). A call to reflect on evaluation practices for failure detection in image classification. In *11th international conference on learning representations (ICLR)* (pp. 1–38). arXiv:2211.15259.

Kaddour, J., Liu, L., Silva, R., & Kusner, M. J. (2022). When do flat minima optimizers work? *Advances in Neural Information Processing Systems, 35*, 16577–16595. arXiv:2202.00661.

Kafunah, J., Ali, M. I., & Breslin, J. G. (2023). Uncertainty-aware ensemble combination method for quality monitoring fault diagnosis in safety-related products. *IEEE Transactions on Industrial Informatics*. https://doi.org/10.1109/TII.2023.3280566

Kull, M., Perello-Nieto, M., Kängsepp, M., Silva Filho, T., Song, H., & Flach, P. (2019). Beyond temperature scaling: Obtaining well-calibrated multiclass probabilities with dirichlet calibration. *Advances in Neural Information Processing Systems, 32*, 12316–12326. arXiv:1910.12656.

Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems, 30*, 6402–6413. arXiv:1612.01474.

Lin, T-Y., Goyal, P., Girshick, R., He, K., & Dollar, P. (2017). Focal loss for dense object detection. In *IEEE international conference on computer vision (iccv)* (pp. 2980–2988). IEEE. https://doi.org/10.1109/iccv.2017.324

Lin, Y.-H., & Li, G.-H. (2022). A bayesian deep learning framework for rul prediction incorporating uncertainty quantification and calibration. *IEEE Transactions on Industrial Informatics, 18*(10), 7274–7284. https://doi.org/10.1109/TII.2022.3156965

Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y. & Guo, B. (2022). Swin transformer v2: Scaling up capacity and resolution. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 12009–12019). IEEE. https://doi.org/10.1109/cvpr52688.2022.01170

Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., & Zhang, G. (2018). Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering, 31*(12), 2346–2363. https://doi.org/10.1109/TKDE.2018.2876857

Maged, A., & Xie, M. (2022). Uncertainty utilization in fault detection using bayesian deep learning. *Journal of Manufacturing Systems, 64*, 316–329. https://doi.org/10.1016/j.jmsy.2022.07.002

Mayr, A., Bauer, J., & Franke, J. (2022). A multi-view deep learning approach for quality assessment in laser welding of hairpin windings based on 2d image captures. *Procedia CIRP, 115*, 196–201. https://doi.org/10.1016/j.procir.2022.10.073

Minderer, M., Djolonga, J., Romijnders, R., Hubis, F., Zhai, X., Houlsby, N., & Lucic, M. (2021). Revisiting the calibration of modern neural networks. *Advances in Neural Information Processing Systems, 34*, 15682–15694. arXiv:2106.07998.

Mishra, R. S., & Ma, Z. Y. (2005). Friction stir welding and processing. *Materials Science and Engineering: R: Reports, 50*(1–2), 1–78. https://doi.org/10.1016/j.mser.2005.07.001

Moon, J., Kim, J., Shin, Y., & Hwang, S. (2020). Confidence-aware learning for deep neural networks. In *37th international conference on machine learning (ICML)* (pp. 7034–7044). PMLR. arXiv:2007.01458.

Mueller, R., Kornblith, S., & Hinton, G. E. (2019). When does label smoothing help? *Advances in Neural Information Processing Systems, 32*, 4694–4703. arXiv:1906.02629.

Mukhoti, J., Kulharia, V., Sanyal, A., Golodetz, S., Torr, P., & Dokania, P. (2020). Calibrating deep neural networks using focal loss. *Advances in Neural Information Processing Systems, 33*, 15288–15299. arXiv:2002.09437.

Naeini, M. P., Cooper, G., & Hauskrecht, M. (2015). Obtaining well calibrated probabilities using bayesian binning. In *29th aaai conference on artificial intelligence* (pp. 2901–2907). AAAI Press. https://doi.org/10.1609/aaai.v29i1.9602

Nwankpa, C., Eze, S., Ijomah, W., Gachagan, A., & Marshall, S. (2021). Achieving remanufacturing inspection using deep learning. *Journal of Remanufacturing, 11*(2), 89–105. https://doi.org/10.1007/s13243-020-00093-9

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., & Snoek, J. (2019). Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. *Advances in Neural Information Processing Systems, 32*, 13991–14002. arXiv:1906.02530.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems, 32*, 8024–8035. arXiv:1912.01703.

Patel, K., Beluch, W. H., Yang, B., Pfeiffer, M., & Zhang, D. (2021). Multi-class uncertainty calibration via mutual information maximization-based binning. In *9th international conference on learning representations (ICLR)* (pp. 1–32). arXiv:2006.13092.

Pyle, R. J., Hughes, R. R., Ali, A. A. S., & Wilcox, P. D. (2022). Uncertainty quantification for deep learning in ultrasonic crack characterization. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, 69*(7), 2339–2351. https://doi.org/10.1109/TUFFC.2022.3176926

Raffin, T., Reichenstein, T., Werner, J., Kühl, A., & Franke, J. (2022). A reference architecture for the operationalization of machine learning models in manufacturing. *Procedia CIRP, 115*, 130–135. https://doi.org/10.1016/j.procir.2022.10.062

Rahimi, A., Shaban, A., Cheng, C.-A., Hartley, R., & Boots, B. (2020). Intra order-preserving functions for calibration of multi-class neural networks. *Advances in Neural Information Processing Systems, 33*, 13456–13467. arXiv:2003.06820.

Rathnakumar, R., Pang, Y., & Liu, Y. (2023). Epistemic and aleatoric uncertainty quantification for crack detection using a bayesian boundary aware convolutional network. *Reliability Engineering & System Safety, 240*, 109547. https://doi.org/10.1016/j.ress.2023.109547

Rožanec, J. M., Bizjak, L., Trajkova, E., Zajec, P., Keizer, J., Fortuna, B., & Mladenić, D. (2023). Active learning and novel model calibration measurements for automated visual inspection in manufacturing. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-023-02098-0

Saiz, F. A., Alfaro, G., & Barandiaran, I. (2021). An inspection and classification system for automotive component remanufacturing industry based on ensemble learning. *Information, 12*(12), 489. https://doi.org/10.3390/info12120489

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2818–2826). IEEE. https://doi.org/10.1109/cvpr.2016.308

Tan, M., & Le, Q.V. (2021). Efficientnetv2: Smaller models and faster training. In *38th international conference on machine learning (ICML)* (pp. 10096–10106). PMLR. arXiv:2104.00298.

Thomas, W. M. (1998). Friction stir welding and related friction process characteristics. In *7th international conference on joints in aluminium (INALCO)* (pp. 157–174).

Tomani, C., Gruber, S., Erdem, M.E., Cremers, D., & Buettner, F. (2021). Post-hoc uncertainty calibration for domain drift scenarios. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 10124–10132). IEEE. https://doi.org/10.1109/cvpr46437.2021.00999

Tomani, C., Waseda, F., Shen, Y., & Cremers, D. (2023). Beyond in-domain scenarios: Robust density-aware calibration. In *40th international conference on machine learning (ICML)* (pp. 34344–34368). PMLR. arXiv:2302.05118.

Vaghefi, E., Hosseini, S., Azimi, M., Shmatok, A., Zhao, R., Prorok, B., & Mirkoohi, E. (2024). Volumetric defect classification in nano-resolution x-ray computed tomography images of laser powder bed fusion via deep learning. *Journal of Manufacturing Processes, 121*, 499–511. https://doi.org/10.1016/j.jmapro.2024.05.030

van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research, 9*(11), 2579–2605.

Vater, J., Pollach, M., Lenz, C., Winkle, D., & Knoll, A. (2021). Quality control and fault classification of laser welded hairpins in electrical motors. In *28th European signal processing conference (EUSIPCO)* (pp. 1377–1381). IEEE. https://doi.org/10.23919/eusipco47968.2020.9287701

Wang, J., Ma, Y., Zhang, L., Gao, R. X., & Wu, D. (2018). Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems, 48*, 144–156. https://doi.org/10.1016/j.jmsy.2018.01.003

Widmer, G., & Kubat, M. (1996). Learning in the presence of concept drift and hidden contexts. *Machine Learning, 23*(1), 69–101. https://doi.org/10.1023/A:1018046501280

Xiao, Y., Shao, H., Feng, M., Han, T., Wan, J., & Liu, B. (2023). Towards trustworthy rotating machinery fault diagnosis via attention uncertainty in transformer. *Journal of Manufacturing Systems, 70*, 186–201. https://doi.org/10.1016/j.jmsy.2023.07.012

Zadrozny, B., & Elkan, C. (2001). Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *18th international conference of machine learning (ICML)* (pp. 609–616). PMLR. https://doi.org/10.5555/645530.655658

Zadrozny, B., & Elkan, C. (2002). Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the 8th acm sigkdd international conference on knowledge discovery and data mining* (pp. 694–699). ACM. https://doi.org/10.1145/775047.775151

Zettler, R., Vugrin, T., & Schmuecker, M. (2010). Effects and defects of friction stir welds. In D. E. A. Lohwasser (Ed.), *Friction stir welding* (pp. 245–276). CRC Press. https://doi.org/10.1533/9781845697716.2.245

Zhang, J., Kailkhura, B., & Han, T.Y-J. (2020). Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. In *37th international conference on machine learning (ICML)* (pp. 11117–11128). PMLR. arXiv:2003.07329.

Zhou, T., Han, T., & Droguett, E. L. (2022). Towards trustworthy machine fault diagnosis: A probabilistic bayesian deep learning framework. *Reliability Engineering & System Safety, 224*, 108525. https://doi.org/10.1016/j.ress.2022.108525

Zhu, F., Cheng, Z., Zhang, X-Y., & Liu, C-L. (2022). Rethinking confidence calibration for failure prediction. In *European conference on computer vision (ECCV)* (pp. 518–536). Springer. https://doi.org/10.1007/978-3-031-19806-9_30