MOXAI – Manufacturing Optimization through Model-Agnostic Explainable AI and Data-Driven Process Tuning

Clemens Heistracher[1]*, Anahid Wachsenegger[2]*, Axel Weißenfeld[2], and Pedro Casas[2]

[1] *craftworks GmbH, Vienna, Austria*
clemens.heistracher@craftworks.at

[2] *AIT Austrian Institute of Technology, Vienna, Austria*
name.surname@ait.ac.at

ABSTRACT

Modern manufacturing equipment offers numerous configurable parameters for optimization, yet operators often underutilize them. Recent advancements in machine learning (ML) have introduced data-driven models in industrial settings, integrating key equipment characteristics. This paper evaluates the performance of ML models in classification tasks, revealing nuanced observations. Understanding model decision-making processes in failure detection is crucial, and a guided approach aids in comprehending model failures, although human verification is essential. We introduce *MOXAI*, a data-driven approach leveraging existing pre-trained ML models to optimize manufacturing machine parameters. MOXAI underscores the significance of explainable artificial intelligence (XAI) in enhancing data-driven process tuning for production optimization and predictive maintenance. MOXAI assists operators in adjusting process settings to mitigate machine failures and production quality degradation, relying on techniques like DiCE for automatic counterfactual generation and LIME to enhance the interpretability of the ML model's decision-making process. Leveraging these two techniques, our research highlights the significance of explaining the model and proposing the recommended parameter setting for improving the process.

1. INTRODUCTION

Today's highly automated manufacturing equipment often provides many configurable parameters to ensure optimal production and accommodate an increased range of products. In practice, machine operators and process engineers rely on a limited set of well-understood key parameters for process controlling and optimization, overlooking the broader space of configurable options and underutilizing the potential to enhance equipment effectiveness. The increased demand for individualization and, consequently, the decrease in batch sizes amplify this effect and further increase the workload for operators. Recent advances in machine learning have led to a surge in data-driven AI/ML models deployed in industrial scenarios for applications such as quality inspection and predictive maintenance, which have integrated key characteristics and patterns of production equipment.

The demand for explainability becomes crucial to optimizing complex manufacturing and production processes as models grow more intricate, resembling "black boxes" that hinder users from understanding the rationale behind predictions. Explainable Artificial Intelligence (XAI) methods address this challenge by providing human-understandable explanations for data-driven decisions. In XAI, two primary categories are evident (Molnar, 2020): model-agnostic and model-specific approaches. Model-agnostic techniques, such as feature importance and surrogate models, offer insights into decision-making processes across various models. Conversely, model-specific methods delve into a model's intrinsic aspects, such as coefficients in linear regression or visualizing decision cuts in decision trees. Local and global scopes characterize explanations, with techniques like Local Interpretable Model-Agnostic (LIME) (Ribeiro, Singh, & Guestrin, 2016), and Shapely Additive Explanations (Lundberg & Lee, 2017) offering local insights. Another popular approach in XAI is counterfactual explanations (Ates, Aksar, Leung, & Coskun, 2021; Jalali, Haslhofer, Kriglstein, & Rauber, 2023), which determine changes to input data necessary for altering a model's output.

In this work, we strive to automate the process of providing recommendations to machine operators in an interpretable manner, empowering them to understand and adjust parameters effectively for optimal performance (Fig. 1). For this purpose, we introduce *MOXAI* tuning, a data-driven approach

leveraging existing pre-trained AI/ML data models to optimize manufacturing machine parameters by applying model-agnostic counterfactual explanations. Given that numerous manufacturing optimization and predictive maintenance tasks are framed within a binary classification –distinguishing between healthy and damaged assets, or regression – predicting health indicators or remaining useful life, counterfactuals emerge as a compelling solution.

To demonstrate the concept and applicability of the proposed approach, we apply MOXAI on the AI4I 2020 Predictive Maintenance Dataset (Matzka, 2020), which is a simulated dataset designed to mirror authentic predictive maintenance data typically observed in industrial manufacturing settings. Applying MOXAI to those samples where the model predicts machine failures, we can analyze the rationale behind these predictions and obtain suggested modifications to fine-tune different process parameters and prevent machine failures. By querying MOXAI for explanations, we assume that mitigating the reasons behind the model failure prediction will result in an enhanced quality outcome. MOXAI explanations are constrained to a subset of features directly or indirectly controlled by the operator. Evaluations involve comparing model suggestions with production settings to quantify the impact on machine failures, by applying LIME to verify the explanations discovered.

The rest of the paper is structured as follows: First, an overview of the related work is given in Section 2 . Section 3 introduces MOXAI. Section 4 describes the experimental setup considered for evaluation purposes, presenting experimental results. Further discussion on results and MOXAI's approach is presented in Section 5. Finally, Section 6 concludes the paper.

2. Related Work

Two main categories emerge in the domain of XAI: model-agnostic and model-specific approaches. Model-agnostic techniques do not rely on specific model characteristics and can generally be applicable across various models to provide insights into their decision-making process (Molnar, 2020). Such methods include feature importance (e.g., shapely values (Lipovetsky & Conklin, 2001)) and model approximation techniques (e.g., surrogate models (Ribeiro et al., 2016)). Conversely, model-specific approaches study the intrinsic aspects of a model and offer a deeper understanding of its learning structure. For instance, coefficients of a linear regression model, visualization of decision cuts of a shallow decision tree, or more complex approaches such as Layer-Wise propagation explanations (Bach et al., 2015), DeepLift (Shrikumar, Greenside, & Kundaje, 2017), and Class Activation Map (Zhou, Khosla, Lapedriza, Oliva, & Torralba, 2016), which visualize the distributed weights of a neural network.

The scope of the explanations provided by either of the aforementioned techniques can be either local (explaining one sample) or global (explaining all the samples) (Molnar, 2020). Local Interpretable Model-Agnostic (LIME) (Ribeiro et al., 2016), and Shapely Additive exPlanations (Lundberg & Lee, 2017) are popular techniques that produce local explanations. Recent studies suggest that the comprehensibility of local explanations, specifically when including the counterfactuals, increases the human understanding of the model's decision boundary (Jalali et al., 2023). Counterfactual explanations are "hypothetical samples that are as similar as possible to the sample that is explained while having a different classification label" (Ates et al., 2021). Therefore, we argue that combining a local explainability approach with generating counterfactuals can help an end user understand the small meaningful changes that cause the shift in the model's decision with minimal computational effort.

Many XAI approaches have been applied in the literature to address manufacturing optimization problems. Schockaert et al. (Schockaert, Macher, & Schmitz, 2020) propose an approach for local interpretability of a model optimized on training data, which forecasts the temperature of the hot metal a blast furnace produces. Combining a Variational AutoEncoder (VAE) with LIME significantly improves generated synthetic samples for training the ML model. Seiffer et al. (Seiffer, Ziekow, Schreier, & Gerling, 2021) develop a framework to detect temporal changes in manufacturing data with SHAP values to enhance error prediction. The framework detects and handles concept drift so that the generated ML models are of sufficient quality in the long term. Jakubowski et al. (Jakubowski, Stanisz, Bobek, & Nalepa, 2021) developed an LSTM autoencoder model for detecting anomalies in the hot rolling process to produce steel coils. They applied SHAP explanations to determine the reasons for anomalies. Regarding model interpretability, Jakubowski et al. (Jakubowski, Stanisz, Bobek, & Nalepa, 2022) employed the SHAP method and counterfactual explanations to gain insight into the decisions made by their trained models. These explanations effectively highlighted the features responsible for the abnormal state of the mill or work rolls, helping identify the anomaly's root cause. Ameli et al. (Ameli et al., 2022) employ XAI methodologies to determine the specific sensors exhibiting anomalies, enhancing decision-making within glass production monitoring. These sensors are localized, analyzing the cause of anomalies by saliency XAI. The approach of Senoner et al. (Senoner, Netland, & Feuerriegel, 2022) involves the development of a data-driven decision model by leveraging high-dimensional data with nonlinear relationships alongside SHAP to discern the intricate relationships between production parameters and manufacturing process quality.

In summary, XAI methods are sporadically utilized in production and predictive maintenance to optimize models and

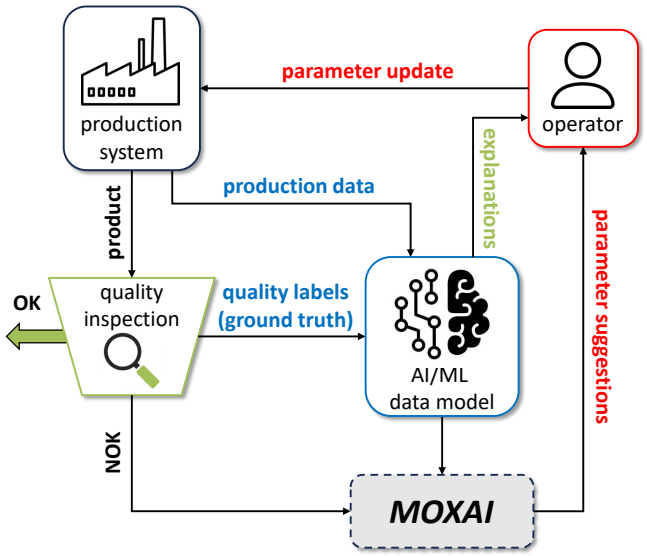

Figure 1. MOXAI information flow for parameter recommendations.

enhance understanding. The approach involving counterfactuals has been minimally employed thus far despite its considerable promise in this domain.

3. Methodology

The general idea of MOXAI is to extract suggestions for *production parameters* (i.e., part of the AI/ML model input features) leading to a desired manufacturing target, relying on pre-trained AI/ML models, and using XAI local explanations through so-called counterfactual instances. A counterfactual instance is a synthetic data point similar to the original instance but with a different model outcome. It is created by perturbing the model's input parameters within certain bounds. Our goal through counterfactual explanations is to answer the question *"what changes to the input (production) parameters of the model would have resulted in a different prediction?"*. MOXAI allows for faster fine-tuning of the configuration of a production process by iterating over instances for which the production output is not as desired.

3.1. Automatic Parameter-Settings' Recommendation

We developed MOXAI to guide machine operators toward better production parameters in case of product quality deviations. We envision a scenario in which process control, quality inspections, or data-driven models indicate a deviation, and the operator is uncertain how to modify the configuration. MOXAI leverages methods from explainable AI to suggest an optimized configuration based on the most recent sample. It requires an existing data model for product quality prediction based on the process parameter and configuration. Our model-agnostic method requires the changeable machine configuration to be part of the model input.

We use the framework for Diverse Counterfactual Explanations (DiCE) (Mothilal, Sharma, & Tan, 2020) to generate recommendations. We leverage counterfactuals to suggest machine parameters that produce good product quality according to the data model. DiCE aims to generate actionable counterfactual sets, ensuring that individual counterfactual examples are feasible and diverse. To achieve this, DiCE adapts diversity metrics through diversity via Determinantal Point Processes (Kulesza, Taskar, et al., 2012) and incorporates feasibility using proximity constraints and user-defined constraints. Process parameters are optimized by extracting the model's capability to determine which parameters lead to a high-quality product. It also addresses sparsity by considering the minimal number of features that must be changed to transition to the counterfactual class. Additionally, it allows users to specify constraints on feature manipulation, such as box constraints on feasible feature ranges, to ensure the practicality of counterfactual examples within real-world constraints. The MOXAI workflow is depicted in Figure 1.

3.2. Human-Guided Correction of Model Failures

To understand the model's decision boundary for detecting defected cases from no-defect cases, we apply LIME, which also offers understandable visualizations for operators and developers to understand why the model failed to detect defective samples. LIME produces instance-based explanations by estimating the decision boundary of the black-box model within a narrow neighborhood. The underlying assumption is that a linear model can effectively approximate the local decision boundary of the black box. The coefficients of this linear model then elucidate the contribution of each feature to the prediction of a sample within this neighborhood. Consequently, LIME's explanations are represented by feature value boundaries. These boundaries signify the impact of each feature; when the feature values fall within these boundaries in a given local neighborhood, they influence the model's decision toward or away from a particular class.

MOXAI's correction algorithm utilizes LIME and examines the top five common explanations provided by this approach for all instances. It performs the following steps: it counts the frequency of these explanations; it then ranks the explanations based on their frequency counts. Next, it records the lowest and highest bounds observed for the most influential feature in the explanations. Human input may be needed from a domain expert who has viewed the data and understands the feature boundaries at this stage. This is necessary because LIME sometimes presents an upper or lower-bound inequality. In such cases, we need to determine the missing boundary. The algorithm then iterates over the generated list and replaces the corresponding feature in the explanations with a randomly generated float within the boundary range. We continue the iteration if this alteration does not rectify the model's prediction. If the alteration corrects the prediction,

Table 1. Results of trained models on the test set.

	Recall	Precision	F1-Score
Nearest Neighbor (KNN)	0.97	0.97	0.97
Decision Tree (DT)	0.99	0.99	0.99
Random Forest (RF)	0.99	0.99	0.99
Gradient Boosting (GBM)	0.99	0.99	0.99
Neural Network (MLP)	0.97	0.94	0.96

Table 2. Detailed results of the trained decision tree on the test set.

	Accuracy	Recall	Precision	F1-score
HDF	0.99	0.95	0.93	0.94
PWF	0.99	0.92	0.89	0.90
OSF	0.99	0.73	0.87	0.78
Machine-Failure	0.99	0.99	0.99	0.99

we proceed to the next misclassified sample. We further emphasize that this approach merely identifies the approximate decision boundary of the model rather than identifying the actual cause of the defect. We can only observe the parameter responsible for the model's misclassification, which may or may not directly correlate with the underlying cause of the defect. The outcomes of this algorithm are discussed in Section 4.3.

4. Experimental Setup and Results

We demonstrate MOXAI's operation using the AI4I dataset, a synthetic dataset commonly used in the scientific community. The AI4I dataset covers a realistic industrial use case and provides an analytical definition for most error types, which can be used to validate corrections as suggested by MOXAI. The dataset consists of 10,000 samples with five numerical features of a milling process, a categorical feature for different product types, and the target variables, which describe the state of five error types:

- Tool wear failure (TWF): The tool fails after a random up-time between 200 - 240 minutes.
- Heat dissipation failure (HDF): The tool fails due to small temperature differences between the tool and air and slow rotational speeds.
- Power failure (PWF): The tool fails for very high or very low power, defined as the product of torque and rotational speed.
- Overstrain failure (OSF): Product variant-dependent error for high tool wear and torque combination.
- Random failures (RNF): A randomly assigned error type.

We exclude TWF and RNF failures from the evaluations due to their random component, as we require an analytical definition of the error for validation.

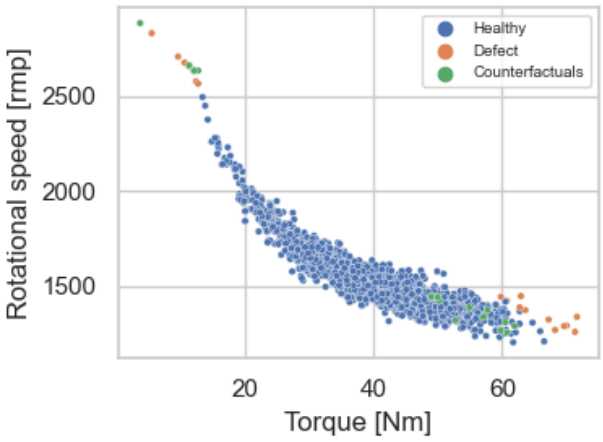

Figure 2. Power Failure (PWF) healthy and defect samples in the test set, as well as generated counterfactual samples, for rotation speed vs. torque of the milling process.

4.1. Data Preprocessing and Modeling

The machine learning model is the core of our approach, and we trained different models following standard best practices. We use a stratified split of 80% of the data for training/validation and 20% for testing, resulting in 7714 samples for the healthy state and 234 defects, of which 115 are HDF, 94 are PWF, and 95 are OSF – note that some machine defects are a combination of multiple failures. The imbalance in the data can be seen as an indication for up-sampling approaches such as SMOTE (Chawla, Bowyer, Hall, & Kegelmeyer, 2002). Still, our experiments showed no significant improvement, and our reported models are trained on the provided data only. We performed experiments using five different model architectures implemented by scikit-learn[1]: k-nearest neighbors , Decision Tree, Random Forest, Gradient Boosting, and Neural Network, and report the results in Table 1, as well as the breakdown of the best-performing model in Table 2, where we see that the detection of the HDF and PWF are more trivial than ODF, which is consistent among the models. Therefore, we can assume that the misclassification of machine failure is potentially caused by detecting the OSF.

4.2. Parameter-Setting Recommendations

MOXAI uses DiCE as an explainer backend, which was initialized using the trained model and the training data. We use the genetic algorithm provided by DiCE, as it supports parameters that prioritize counterfactuals similar to training data and thus avoid regions in the parameter space that are not well defined due to missing training data. We allow variation in all features, but real-life use cases will likely require limiting the parameters that can be modified at the machine.

[1] https://scikit-learn.org

Table 3. Accuracy of suggested parameters.

	KNN	DT	RF	GBM	MLP
HDF	0.89	1.0	1.0	0.94	0.78
PWF	0.91	1.0	1.0	1.0	0.71
OSF	0.58	0.95	0.95	0.84	0.47
overall_accuracy	0.81	0.98	0.98	0.93	0.68

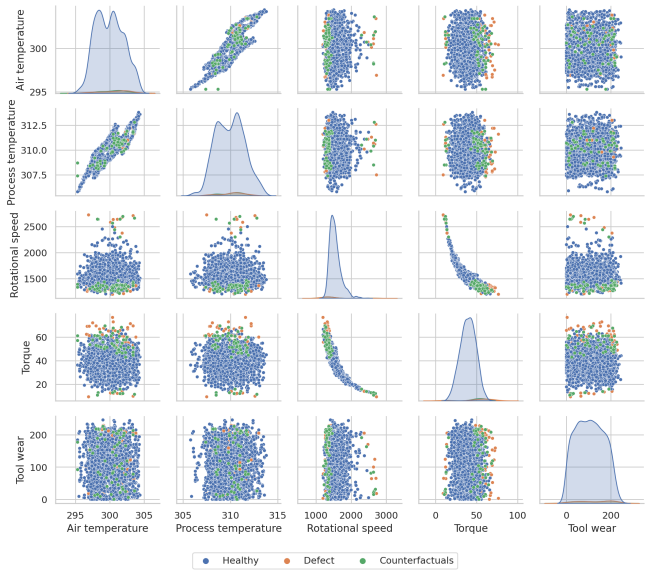

Figure 3. Pairplot of the test set and created counterfactual samples, for the five different numerical features characterizing the milling process.

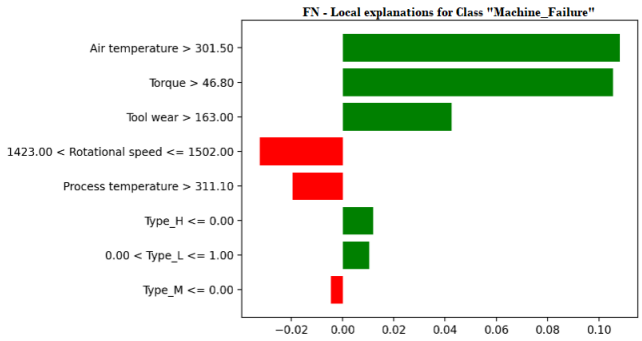

Figure 4. LIME's local explanations for a misclassified sample as not a machine failure (FN).

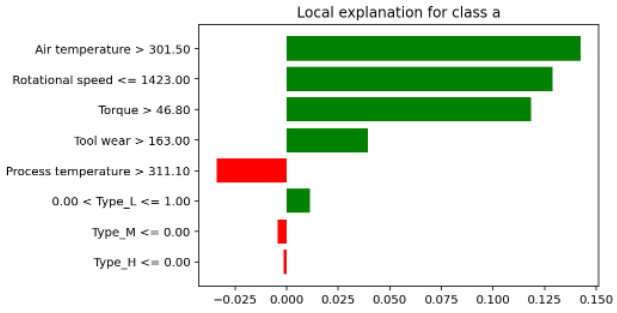

Figure 5. LIME's local explanations for a sample correctly classified as a machine failure (TP).

To evaluate MOXAI, we use the analytic definition of errors provided by the AI4I dataset creators. For each defective sample of the test set, we use MOXAI to calculate a suggested set of machine parameters. Figures 2 and 3 depict the healthy and defective samples in the test set and the generated counterfactuals. We take the error definition to determine if the solution proposed by MOXAI actually solved the problem and corresponded to a healthy product. We report the percentage of successful corrections as accuracy in Table 3.

4.3. Correction of Model Failures

We generate LIME explanations for each failed sample using the models discussed in the preceding section. We encounter 21 failed samples, comprising 15 false negatives (FNs) and six false positives (FPs). Through the analysis of modeling separated failure modes, we noticed that these misclassifications predominantly stem from the model's failed attempt to detect PFW and OSF accurately. We extract the explanations using the algorithm detailed in Section 3.2. To correct false positives (FPs), we randomly generate float values within the approximate feature range identified by LIME to produce counterfactual instances. We leverage our understanding of value ranges given by dataset providers, contributing to failures in each specific mode, to rectify errors in parameter settings. Similarly, we apply this method to false negatives (FNs), mostly from inaccuracies in process temperature values. By generating counterfactual instances, we illustrate the adjustments required in parameter values to identify defective samples accurately. In Figure 4 and Figure 5, we demonstrate a comparison of a true positive (correctly detected machine failure) with a scenario where the model predicted a failure as "not a failure" with low confidence (the prediction probability for the class Failure is 0.47) explained by LIME. The plot shows that, even though the features *Air temperature, Torque* and *Tool wear* are positively contributing to this prediction being a failed sample, the values of *Rotational speed* and *Process temperature* are shifting the model's decision towards the class "not a Failure". MOXAI suggests a minor change of *Process Temperature* to a value slightly smaller than 311.10, creates a counterfactual, and corrects this prediction. In practice, the domain expert should verify whether this change is valid and does not contradict the definition of this failure mode.

5. Discussion

The evaluation of model performance in a classification task unveils nuanced observations. While all models exhibit satisfactory accuracy in data classification, their effectiveness

in generating reliable counterfactual samples varies. Notably, tree-based models emerge as the most robust, surpassing alternative methodologies, such as Multi-Layer Perceptron (MLP) and K-Nearest Neighbors (KNN). Furthermore, an analysis of error types reveals differences among model performances. Specifically, most models demonstrate proficiency in addressing tool wear (PWF) and heat dissipation (HDF) errors but struggle when confronted with errors arising from multiple product types (OSF). These findings underscore the importance of assessing classification accuracy and considering models' ability to provide dependable counterfactual samples and their efficacy in handling diverse error types. Moreover, the current state of MOXAI is limited to the parameters within the proximity of its training set. Therefore, it cannot suggest optimizations for unseen production scenarios. One approach to address this limitation could be the usage of digital twin solutions that are more flexible when it comes to approximating new parameters and production settings.

We underscore the significance of comprehending the model's decision-making process in failure detection and why these particular counterfactuals were suggested. A guided approach aids in understanding why a model failed and whether the model's identified correlations are logical. While MOXAI offers an interpretable and human-in-the-loop system for comprehending model failures and suggesting meaningful samples tailored to this specific use case, the semi-automatic counterfactuals produced by our human-guided approach could benefit from considering feature co-linearities and interactions, and a domain expert should verify them to exclude nonsensical examples. This process is crucial for gauging the model's reliability and assessing the suitability of a fully automated counterfactual generation module. Therefore, the operator can plainly trust the model's recommendations to choose the best settings based on the explanations provided by LIME's output.

6. Conclusion

The approach of XAI to enhance data-driven process tuning for optimizing production or predictive maintenance is promising. MOXAI proposes a data-driven, XAI-powered approach to optimizing manufacturing machine parameters, relying on pre-trained ML models of any nature. We have trained different ML models for failure prediction in a popular synthetic dataset representing a realistic industrial scenario, applying MOXAI's information flow to identify potential corrections to improve failure samples and improve understanding of the operation of these ML models.

DiCE is a key element in automatically generating counterfactual explanations, which can assist operators in adjusting process settings so that machine failures or degraded production quality can be reduced. Applying LIME explanations to address false predictions within our model proved insightful. We successfully rectified both false positives and false negatives by analyzing failure modes and generating counterfactual instances based on LIME insights. Additionally, our demonstration of LIME's output underscores its potential to enhance model decisions' interpretability.

Enhancing the understanding of counterfactual methods is important for future advancements. This ensures that such methods foster a causal understanding for human operators while avoiding any risks of biased, sub-optimal, or erroneous explanations.

Acknowledgments

This work has been funded by the Austrian Research Promotion Agency (FFG) under grant No. 883864 *ZDM – Zero Defect Manufacturing* and grant No. FO999913202 *UNDERPIN* and by the European Commission under contract No. 101123179 *UNDERPIN*.

References

Ameli, M., Becker, P. A., Lankers, K., van Ackeren, M., Bähring, H., & Maaß, W. (2022). Explainable unsupervised multi-sensor industrial anomaly detection and categorization. In *2022 21st ieee international conference on machine learning and applications (icmla)* (pp. 1468–1475).

Ates, E., Aksar, B., Leung, V. J., & Coskun, A. K. (2021). Counterfactual explanations for multivariate time series. In *2021 international conference on applied artificial intelligence (icapai)* (pp. 1–8).

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, *10*(7), e0130140.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321–357.

Jakubowski, J., Stanisz, P., Bobek, S., & Nalepa, G. J. (2021). Explainable anomaly detection for hot-rolling industrial process. In *2021 ieee 8th international conference on data science and advanced analytics (dsaa)* (pp. 1–10).

Jakubowski, J., Stanisz, P., Bobek, S., & Nalepa, G. J. (2022). Roll wear prediction in strip cold rolling with physics-informed autoencoder and counterfactual explanations. In *2022 ieee 9th international conference on data science and advanced analytics (dsaa)* (p. 1-10). doi: 10.1109/DSAA54385.2022.10032357

Jalali, A., Haslhofer, B., Kriglstein, S., & Rauber, A. (2023). Predictability and comprehensibility in post-hoc xai methods: A user-centered analysis. In *Science and in-*

formation conference (pp. 712–733).

Kulesza, A., Taskar, B., et al. (2012). Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, *5*(2–3), 123–286.

Lipovetsky, S., & Conklin, M. (2001). Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, *17*(4), 319-330. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1002/asmb.446 doi: https://doi.org/10.1002/asmb.446

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In I. Guyon et al. (Eds.), *Advances in neural information processing systems* (Vol. 30). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf

Matzka, S. (2020). Explainable artificial intelligence for predictive maintenance applications. In *2020 third international conference on artificial intelligence for industries (ai4i)* (pp. 69–74).

Molnar, C. (2020). *Interpretable machine learning*. Lulu.com.

Mothilal, R. K., Sharma, A., & Tan, C. (2020). Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 conference on fairness, accountability, and transparency* (p. 607–617). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3351095.3372850 doi: 10.1145/3351095.3372850

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (p. 1135–1144). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/2939672.2939778 doi: 10.1145/2939672.2939778

Schockaert, C., Macher, V., & Schmitz, A. (2020). Vae-lime: deep generative model based approach for local data-driven model interpretability applied to the ironmaking industry. *arXiv preprint arXiv:2007.10256*.

Seiffer, C., Ziekow, H., Schreier, U., & Gerling, A. (2021). Detection of concept drift in manufacturing data with shap values to improve error prediction. In *Data analytics 2021: The tenth international conference on data analytics* (pp. 51–60).

Senoner, J., Netland, T., & Feuerriegel, S. (2022). Using explainable artificial intelligence to improve process quality: Evidence from semiconductor manufacturing. *Management Science*, *68*(8), 5704–5723.

Shrikumar, A., Greenside, P., & Kundaje, A. (2017). Learning important features through propagating activation differences. In *Proceedings of the 34th international conference on machine learning - volume 70* (p. 3145–3153). JMLR.org.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning deep features for discriminative localization. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 2921–2929).