Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 138 (2026) 1015–1020

www.elsevier.com/locate/procedia

18th CIRP Conference on Intelligent Computation in Manufacturing Engineering

Real-time Quality Monitoring of Wire EDM Process Using Machine Learning

Akshay Paranjape[a]*, Ugur Küpper[b], Daniel Schulze Brock[b], Robert H. Schmitt[c], Markus Ohlenforst[a], Martin Peterek[a]

[a] *IconPro GmbH, Friedlandstraße 18, 52064 Aachen, Germany*
[b] *Manufacturing Technology Institute (MTI) of RWTH Aachen University, Campus-Boulevard 30, 52074 Aachen, Germany*
[c] *Laboratory for Machne Tools and Production Engineering (WZL) of RWTH Aachen University, Campus-Boulevard 30, 52074 Aachen, Germany*

* Corresponding author. Tel.: +49-241-41259946, *E-mail address:* akshay.paranjape@iconpro.com

Abstract

Traditional quality assessment in EDM involves labor-intensive manual evaluation of geometric deviations. Anticipating curvatures post-main cut allows compensation through trim cut adaptation, streamlining the process. This paper presents a novel approach for automating and real-time quality monitoring of Wire EDM processes through the integration of machine learning models trained on historical process and quality data. The effectiveness of this method is validated on industrial Wire EDM machine along with feature importance analysis to provide insights into the underlying patterns captured. Testing on industrial unseen data demonstrates a mean absolute error of approx. 1 µm for geometric curvature deviation in the Wire EDM process.

© 2025 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the 18th CIRP Conference on Intelligent Computation in Manufacturing Engineering (CIRP ICME '24)

Keywords: Wire EDM, Real-time Monitoring, Predictive Quality, Machine Learning;

1. Introduction

Wire Electric discharge machine (wire-EDM) is widely used for fine precision cutting mainly in the aerospace industry. It is often used for machining intricate and complex shapes with high tolerances. To ensure the precision of wire EDM, a physical measurement of the accuracy of the cut must be quantified. This is done with geometric curvature error measurements, mostly with coordinate measurement machines (CMMs). This entails substantial time and expense, along with a significant requirement for manual labor. The conducted research explores the possibility of detection of geometric curvature to ensure real-time quality monitoring of the wire-EDM process with a data-driven approach. An automated loop for monitoring is proposed with machine learning (ML) based models. These models are block-box in their inherent nature. An interpretation of the trained ML models is provided to draw relevant process insights from the analysis. An end-to-end pipeline of ML lifecycle for wire EDM process is proposed, combining intermediate steps such as data generation, feature generation, modelling, ensemble creation and model deployment.

Wire EDM is a machining process that continuously erodes material through a series of electric discharges between the wire electrode and the workpiece. The workpiece is a metal or an alloy that is to be cut. The wire is guided close to the workpiece and its position is controlled with CNC technology. A huge amount of data is generated during the complete process.

2212-8271 © 2025 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the 18th CIRP Conference on Intelligent Computation in Manufacturing Engineering (CIRP ICME '24)
10.1016/j.procir.2026.01.175

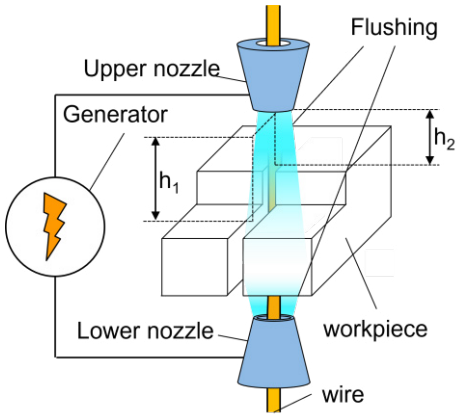


Fig. 1 Wire EDM experimental set up design

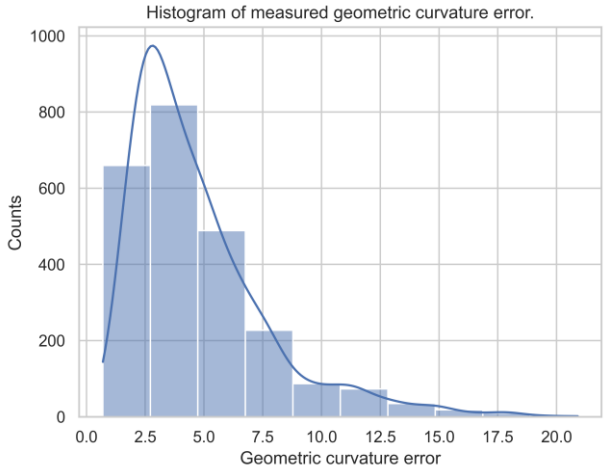


Fig. 2 Histogram of geometric curvature error measurements

Küpper et al. [1] discuss the digitalization of the wire EDM process. The very first usage of machine learning techniques for wire-EDM was demonstrated by Liao et al. in their attempt to estimate the workpiece height in Wire EDM [2]. To maintain machining speed for various heights, it is important to know the height of the workpiece. The machining parameters can then be derived from the height of the workpiece. Successful estimation of workpiece height with an accuracy of 1.6 mm is demonstrated by Liao et al. [2]. Caggiano et al. explored monitoring of the Wire EDM process through advanced sensor signal processing [3]. Through the monitoring system, they identified operating conditions responsible for the generation of defects like lines and marks on the final workpiece surface. Kupper et al. explored the possibility of real-time process monitoring of the Wire EDM process by recording and analyzing discharges and classifying them as normal and short circuits. The study was limited to the usage of statistical techniques to find the correlations and was successful in deriving rules for monitoring the process [4]. Material removal in the Wire EDM process directly impacts the quality of the resulting cut surface. A methodology and system were developed by Kupper et al. to visualize spatially resolved process energy in wire EDM [5]. They further facilitated the mapping of generated process energy on workpiece height. Data-driven models for the prediction of geometric accuracy of machined workpieces have been previously developed by analyzing process data [1]. This is one of the first steps toward digitalization of the Wire EDM process. In the context of predictive quality in the manufacturing environment, Mueller et al. [6] described a fully automated system. This paper advances the research to new stages for wire EDM process. To the best of our knowledge, no research has advanced with an end-to-end machine learning pipeline for real-time prediction of geometric accuracy of machined workpieces through the wire-EDM process.

2. Data generation

Data generation is the first step in the process of exploration of data-driven methods. The ML algorithms can only be as good as the data [7]. The experimental setup for the data generation for the wire EDM machine is shown in Fig. 1. The cutting takes place in the X-direction and the workpiece is mounted along the Z-axis. The necessary data required for ML modelling was generated in a series of experiments where workpiece parts with a length of l = 30 mm were cut and the corresponding process characteristics and resulting workpiece curvature were measured. The steel *X155CrVMo12-1* was used as the workpiece. It is a high-quality tool steel and can be eroded very well. The tests were carried out on the Makino U6 H.E.A.T. This wire EDM machine is frequently used in industry with deionized water as the dielectric. The wires that were used had a diameter of d = 0.25 mm. To generate a versatile data basis for the model, various process conditions were artificially generated due to changes in the process influencing factors. These generate different process characteristics, resulting in different surface curvatures. The different process conditions are realized by various rinsing conditions, workpiece heights, wire materials, and machine settings. Variation in rinsing conditions is achieved by altering the separation between the upper and lower rinsing nozzles relative to the workpiece. The distances a = 10, 50 and 100 mm between the workpiece and nozzle were used. This reduces the flushing pressure within the working gap. Varying workpiece heights can also have a major influence on the rinsing conditions and process behavior. Workpieces with heights of h = 40, 60, 80 and 120 mm were used in the tests. Furthermore, different wires were used for the experiments. Uncoated brass wires (bercocut pro, better brass) and coated wires (topas H.E.A.T.) were used. In addition to these process influencing factors, many settings in the machine parameters were changed. Machine parameters in connection with the unloading energy have a particularly large influence on the contour deviation. Therefore, the parameters of discharge duration (*a* & *b* for normal discharge; *c* & *d* for short circuit discharge), pause duration (*off*) and the maximum current of a discharge *(ipm*) were varied. In addition, different variants of the servo reference voltage (*SV_ADJ*) were used, and the flushing pressure was changed. For the tests, the machine settings mentioned were increased or reduced based on a standard technology, so that there were different levels for each parameter. Two new parameters were engineered by combining discharge duration and pause duration (*factor_1* & *factor_2*).

The electrical signals resulting from these experiments cannot be stored as continuous current signals due to the high sampling rate required for this process and the associated high volume of data. Therefore, the measurement system developed by Küpper et al. was used to describe the continuous process character. This detects the process character derived from the process conditions by recording the current signals and analysing it in real time. The high-performance FPGA hardware used generates characteristic values for each discharge, which can be used to characterise the hole process [4]. Furthermore, the disparity in current between the lower and upper wire guides can determine the discharge position along the wire. By incorporating wire position data from the machine control, each discharge can be precisely mapped to a specific location on the workpiece surface [5].

One single discharge has a negligible influence on the processing. Only through the high-frequency energy input into the workpiece is a measurable removal generated, so the sum of the individual discharges should be considered for the modelling. The measurement technology allows an exact, spatially resolved determination of the discharges. For this purpose, the machined surface is divided into h = 1 mm high and w = 1 mm wide areas, and new process parameters are defined for these, which describe the process character of the discharges depending on the location. This enables a significant reduction in data, as millions of pieces of data can be represented by a few new process parameters. The formation of the new characteristic values was realised using a MATLAB script. These include number of discharges, discharge frequency, feed rate, energy input, average ignition delay time, number of consecutive discharges, proportion of standard discharge, average discharge current, and E-factor. The E-factor is a statistical factor that considers the discharge distribution along the wire electrode. Küpper et al. were able to prove the influence of this value on the curvature in his investigations so that this influential factor should be included in the modelling [1].

With regards to all the data generation steps mentioned above, a dataset of size 2400 samples was generated. Each workpiece was analysed using a CMM measuring device for geometric curvature accuracy. The variation in geometric curvature errors is depicted in Fig. 2. Section 3 discusses in detail the modelling aspects of machine learning for the prediction of geometric accuracy.

3. Machine learning modelling

After preprocessing, the number of features or input parameters was reduced to 25. This dataset was then split into train, validation, and test split in a 70-15-15 ratio. Training and validation splits were used for training and fine-tuning the model whereas all the metrics are reported on the test split. This is the common practice followed for ML model training.

3.1. Model training

Multiple machine learning models were trained to predict geometric curvature errors. Instead of depending on individually trained models, a diverse ensemble of models was developed since the ensemble model tends to perform better than individual models [8]. Since the target variable, geometric curvature error is numerical, regression models were employed for training. These include a combination of simple tree-based models, decision trees, random forests, boosted trees, and neural network models. The training for these models was performed in an automated fashion using a runtime of 60 minutes. These models were trained and fine-tuned on training and validation datasets respectively. Insight from Gijsbers at al.[9] were used to train the individual model on open-source libraries. These models were later combined into an ensemble (cf. Section 4.2). Fig. 2 illustrates the training loop mechanism with various components from data collection to model evaluation. EDM process data along with quality measurements with CMMs constitute the training dataset. Custom feature extractor module is used to extract features from raw generated data. Train and validation set is used to train an ensemble of ML model which are finally evaluated on test split.

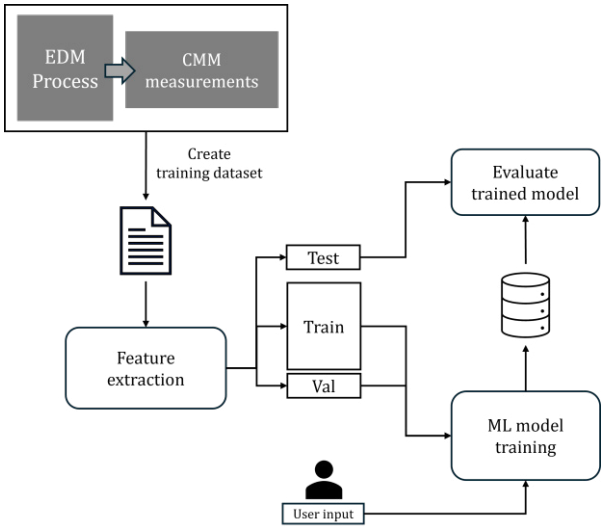


Fig. 3 Training loop mechanism

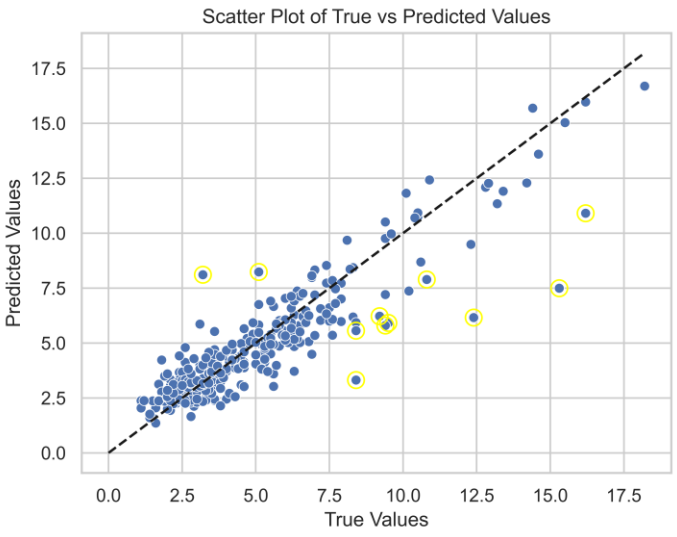


Fig. 4 Scatter plot of true and the predicted values (unit µm).

3.2. Model ensemble

The trained models were grouped to provide the highest accuracy. Thus, the resultant global model is not a single model but rather an ensemble of models. The ensemble is created by a voting mechanism. The voting mechanism combines the prediction of multiple individual models and produces a final prediction by averaging the predictions for a regression task. A total of 108 models were included in the final ensemble. These models can be categorized into major categories as described in Table 1. The overall ensemble is a combination of neural networks and various decision tree-based machine learning algorithms. Default hyperparameters were used for training all the algorithms for the given runtime. In the ensemble creation, individual models can receive multiple votes. The training was carried out in Python using open-source libraries. The creation of an ensemble with a voting mechanism is also theoretical proof that ensures that the ensemble performs better than the individual models.

Table 1 Categorized model of the final ML ensemble model.

Model category	No. of votes
Neural Network	45
CatBoost	20
LightGBM	17
XGBoost	10
RandomForests	8
ExtraTrees	8

3.3. Evaluation of the trained model

The global ensemble model was evaluated on test split since this dataset is completely unseen by the model or used by the model for hyperparameter tuning. Fig. 3 shows an illustration of a scatter plot with true curvature values on the x-axis and predicted curvature values on the y-axis. A 45-degree line implies perfect prediction for an ideal model, the deviation of prediction values from the actual value can be compared with the 45-degree line. Quantitative estimation of the predicted values (cf. Table 2) is done using metrics R2-score (0.82), mean absolute error (0.87 µm), and root mean squared error (1.29 µm). The evaluated error on test split was under the accepted limit for the use-case. Fig. 3 shows that very few samples have mean absolute error higher than 2 µm. The predicted values with MAE higher than 2 µm are highlighted in yellow.

To confirm any bias associated with the splitting of the dataset, four experiments were performed with unique random seeds for splitting the dataset. The R2-score for these experiments was 0.81, 0.82, 0.79, and 0.82. This concludes that there is negligible bias in the results for the trained model described in previous sub-sections. The ensemble composition for these four experiments differs slightly as compared to Table 1 but it maintains the evaluated R2-score close the 0.80.

Table 2 Trained ensemble model metrics on the test split.

Metrics	Score
R2-score	0.82
Mean absolute error	0.87
Root mean squared error	1.67

3.4. Model interpretation

Statistical models like linear regressor, rule-based decision, etc. are up to some extent interpretable. A multidimensional linear regression formula can be interpreted for the importance of the input parameters based on the weight of that parameter in the equation. A similar interpretation for the machine learning model is not directly possible because of its non-linear nature.

Permutation feature importance proposed by Leo Breiman [10] where each input feature is permuted *n* times to calculate the effect on the target is among the first frameworks for model interpretation. Open source Random Forest algorithm comes up with an inbuilt mechanism for calculating these importances. However, this technique may not adequately address correlated features, leading to potentially biased or unstable importance scores. Lundberg et al. [11] introduced the concept of Shapley values from Game theory to assess the impact of input parameters on the target value. They applied Shapley values, known by the acronym SHAP (Shapley Additive exPlanation), to model the influence of input values

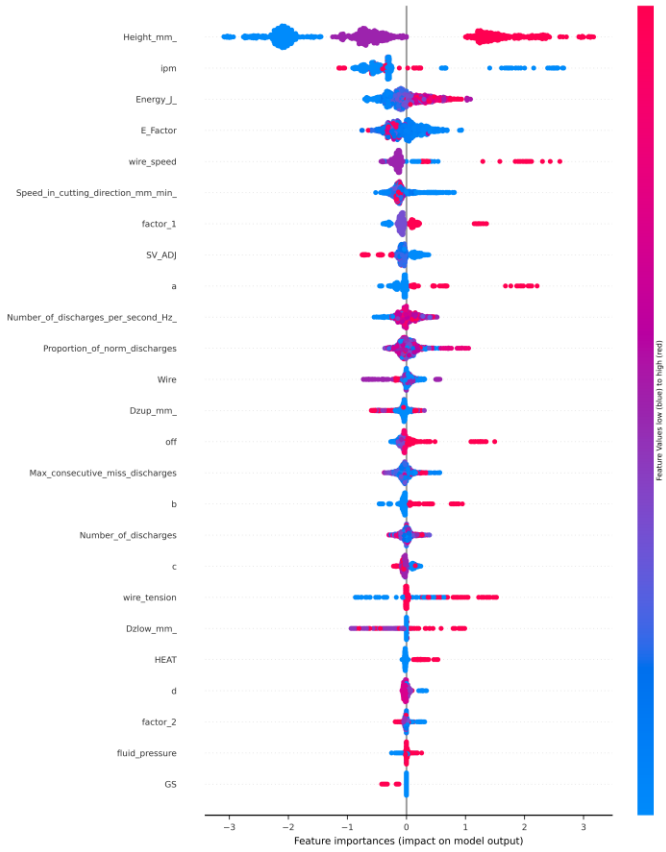


Fig. 5 Model interpretation with SHAP values with feature-wise histogram.



Fig. 6 Influence of various process parameters that lead to geometric curvature error of 19.

on the model prediction. Ideal calculation of shapley regression values requires retraining of the model with and without a particular feature. The predictions from these two models are compared on the input parameters. To avoid re-training of the model, Lundberg et al. utilized the concept of background data that creates a pseudo model with feature subset S of all the features F. Eq. (1) shows the calculation of Shapley values for feature *i*. The effect of the feature is calculated as a difference of input feature vector x_S with and without the feature *i.* This is repeated for every subset of *S of all features F.*

$$\phi_i = \sum_{S \subset F\{i\}} \frac{|S|!(|F|-|S|-1)!}{|F|!} \left[f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S) \right] \quad (1)$$

SHAP implementation utilizes background data to establish a baseline for comparison when evaluating the impact of various features.

In Fig. 5, the x-axis represents SHAP values in the same scale as the target parameter, and the y-axis represents the histogram for the calculated feature. The feature values are color coded between blue (low) and red (high). The plotted values can be interpreted as the value contributed by each feature to reach the final numerical target value. According to the calculated values, the height of the workpiece has the highest influence on the curvature. The color-coded scatter plot of height implies that the lower height of the sample tends to have lower geometric curvature error and higher height samples tend to have higher geometric curvature. A similar interpretation can be drawn from other features of the plot. Dissipated energy, cutting speed, per second discharge frequency, E-factor, and wire speed are among the top 30% important features. On average a higher wire speed leads to high geometric curvature error. Fluid pressure has negligible importance for decision making. Heat technology (*HEAT*) and cutting strategy (*GS*) seem to have a low impact on the final prediction. The machine parameter *ipm* has a very impact with second rank in the order of importance. Similarly other features *factor1*, *SV_ADJ* and *a* show high importance. From the plot, it can be concluded that higher values of *factor1* and *a* lead to high geometrics errors. The relations are opposite for *SV_ADJ.* The zero point on the graph is for the background data. The *wire* type is not among the top influencing features.

Initial insight for the optimization of the process can be drawn with the SHAP beeswarm plot (Fig. 5). However, it is to be noted that the conclusions from the plot are on a global level and may specifically comply with every experiment. It is evident from Fig. 3 that the dataset has samples with higher geometric curvature errors up to 20. One such example was analyzed for high curvature error with the SHAP force plot in Fig. 6. This shows the individual contribution of the feature to reach to final prediction value of 19. The base value on the plot represents the average prediction for the background data used by SHAP explainer.

3.5. Model deployment

For usage in the industrial context, the trained machine learning model needs to be deployed to make it accessible for the end users. The overall deployment is done under a Kubernetes cluster to leverage the features of Kubernetes like non-breaking, auto-scaling, resource management, and production readiness [12]. The training of the model is conducted via a Kubernetes job inside the Kubernetes cluster. Once the model is trained it is stored on S3 storage. The S3 storage was used because of its simple and efficient storage architecture. Using a flask server, the model is deployed in a pod. The connection of the Kubernetes cluster to the outside world takes place via an ingress controller for both training and inference. The ingress resource facilitates the routing of the inference request toward the inference pod within the Kubernetes cluster. The response to the predicted curvature error is received as a JSON output. The JSON output can then be used to display on the end-user dashboard.

The deployed model was tested against the runtime in an industrial setup. The average runtime for the entire end-to-end pipeline was measured at 2 seconds per prediction request. This runtime remains relatively consistent across sample sizes ranging from 1 to 200. However, it escalates to 3 seconds for a batch of 300 samples and further extends to 4.5 seconds for a batch of 1000 samples. The deployed model is intended to be used in the industrial context to potentially reduce the cost and effort for manual measurement for the parts cut with wire-EDM. The inspection cost can be reduced by inspecting only those parts that are prone to higher geometric curvature values before classifying them as scrap parts.

4. Summary and Outlook

The research conducted employs an end-to-end pipeline for predictive quality evaluation for the wire-EDM process including machine learning model selection, model training, model interpretation, and model deployment. A brief about

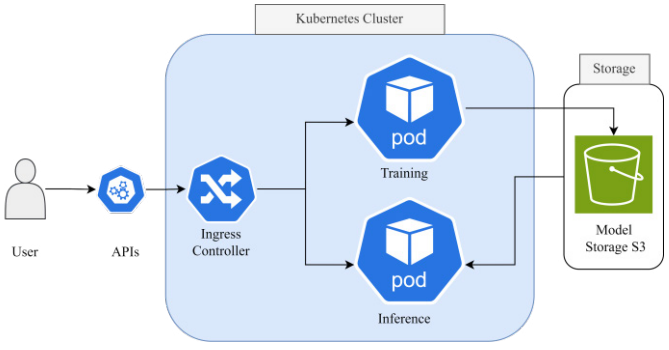


Fig. 7 Training and deployment architecture under Kubernetes orchestration.

industrial usage shows that the model is compatible with production usage. With the use of the detailed pipeline, a reduction in the inspection cost can be achieved. The paper intends to provide data-driven advancement in wire-EDM. Prior work in the field focuses on preliminary data-driven analysis and is far off from the complete end-to-end user pipeline. The main results of the paper are summarized below:

- Trained ML model could predict geometric curvature error with mean absolute error approx.1 µm.
- Reported runtime of 2 seconds per 100 predictions in an industrial environment for predicting geometric accuracy.
- Along with process parameters, machining parameters also have a good influence on target predictions as derived from model interpretation.
- Other than the height of the cut piece, the *Dissipated Energy* and *E-factor* (ratio of short circuit to normal discharge) play a major role in determining curvature error.
- Proposed an end-to-end ML lifecycle framework for wire-EDM to facilitate the development of an industrial application aimed at advancing data-driven decision-making.

The authors intend to further research in the direction of technology development by optimizing the machining parameters for achieving higher cutting speed for wire-EDM while at the same time maintaining or further enhancing the quality of the process.

Acknowledgements

The authors would like to acknowledge the German Federal Ministry of Education and Research (BMBF) for funding project “KI-Erosion” (project number 01IS20096). The authors are responsible for the contents.

References

[1] U. Küpper, E. Tchoupe, A. Klink, and T. Bergs, “Prediction of Geometrical Accuracy in Wire EDM by Analyzing Process Data,” in Procedia CIRP, Elsevier B.V., 2022, pp. 23–28. doi: 10.1016/j.procir.2022.09.114.

[2] Y. S. Liao, M. T. Yan, and C. C. Chang, “A neural network approach for the on-line estimation of workpiece height in WEDM”.

[3] A. Caggiano, R. Perez, T. Segreto, R. Teti, and P. Xirouchakis, “Advanced Sensor Signal Feature Extraction and Pattern Recognition for Wire EDM Process Monitoring,” Procedia CIRP, vol. 42, pp. 34–39, Jan. 2016, doi: 10.1016/J.PROCIR.2016.02.181.

[4] U. Küpper, T. Herrig, A. Klink, D. Welling, and T. Bergs, “Evaluation of the Process Performance in Wire EDM Based on an Online Process Monitoring System,” Procedia CIRP, vol. 95, pp. 360–365, Jan. 2020, doi: 10.1016/J.PROCIR.2020.02.325.

[5] U. Küpper, T. Herrig, and T. Bergs, “Visualization of Spatially Resolved Energy in Wire Electrical Discharge Machining,” in Procedia CIRP, Elsevier B.V., 2021, pp. 1512–1517. doi: 10.1016/j.procir.2021.11.255.

[6] T. Mueller, R. Kiesel, and R. H. Schmitt, “Automated and Predictive Risk Assessment in Modern Manufacturing Based on Machine Learning,” Advances in Production Research, pp. 91–100, 2019, doi: 10.1007/978-3-030-03451-1_10.

[7] L. Leyendecker, M. Zuric, M. A. Nazar, K. Johannes, and R. H. Schmitt, “Predictive Quality Modeling for Ultra-Short-Pulse Laser Structuring utilizing Machine Learning,” Procedia CIRP, vol. 117, pp. 275–280, 2023, doi: 10.1016/J.PROCIR.2023.03.047.

[8] A. Mohammed and R. Kora, “A comprehensive review on ensemble deep learning: Opportunities and challenges,” 2023, doi: 10.1016/j.jksuci.2023.01.014.

[9] P. Gijsbers et al., “AMLB: an AutoML Benchmark,” Journal of Machine Learning Research, vol. 1, pp. 1–48, 2000, Accessed: Apr. 05, 2024. [Online]. Available: https://www.acm.org/publications/policies/artifact-review-and-badging-current

[10] L. Breiman, “Random Forests,” 2001, Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324.

[11] S. Lundberg and S.-I. Lee, “A Unified Approach to Interpreting Model Predictions,” May 2017, [Online]. Available: http://arxiv.org/abs/1705.07874

[12] C. Carrión, “Kubernetes as a Standard Container Orchestrator - A Bibliometric Analysis,” J Grid Comput, vol. 20, no. 4, pp. 1–23, Dec. 2022, doi: 10.1007/S10723-022-09629-8/METRICS.