Production Engineering (2023) 17:237–245
https://doi.org/10.1007/s11740-022-01173-4

MECHANICAL ENGINEERING

Quality prediction for milling processes: automated parametrization of an end-to-end machine learning pipeline

Alexander Fertig[1] · Christoph Preis[1] · Matthias Weigold[1]

Received: 13 September 2022 / Accepted: 21 November 2022 / Published online: 29 November 2022

Abstract

The application of modern edge computing solutions within machine tools increasingly empowers the recording and further processing of internal data streams. The datasets derived by contextualized data acquisition form the basis for the development of novel data-driven approaches for quality monitoring. Nevertheless, for the desired data-driven modeling and data handling, heavily specialized human resources are required. Additionally, domain experts are indispensable for adequate data preparation. To reduce the manual effort regarding data analysis and modeling this paper presents a new approach for an automated parametrization of an end-to-end machine learning pipeline (MLPL) to develop and select the best-performing quality prediction models for usage in machining production. This supports domain experts with a lack of specific knowledge of data science to develop well-performing models for machine learning-based quality prediction of milled workpieces. The results show that the presented algorithm enables the automated generation of data-driven models at high prediction performances to use for quality monitoring systems. The algorithm's performance is tested and evaluated on four real-world datasets to ensure transferability.

Keywords Quality prediction · Machine learning · Milling · Machine tool data

1 Introduction

In production with machine tools it is increasingly possible to record and process the internally processed data at high frequencies. By means of context-sensitive data acquisition [11], it is becoming feasible to automatically evaluate the obtained high-quality datasets and to utilize these for developing new data-driven approaches to process optimization and quality monitoring. Nevertheless, for the desired data-driven modeling and data handling, heavily specialized human resources are required [24]. Therefore, it is important to use the domain knowledge of experts for adequate data preparation and to automate the subsequent development of ML-based predictive models as well as possible.

Based on the results from Fertig et al. [12] this paper presents a new approach for an automated parametrization of an end-to-end machine learning pipeline (MLPL) to develop quality prediction models for usage in machining production. The presented algorithm provides an individual identification and parameterization of appropriate methods for feature extraction and selection. The obtained findings are used to build models for the prediction of the manufactured workpiece quality. This enables domain experts with a lack of specific knowledge of data science to develop automatically predictive models based on an acquired production dataset. Further these models can be used in machine learning based quality monitoring systems.

2 State of the art

In the field of quality prediction of machined workpieces, there are some research publications, which, as shown in Fertig et al. [12], can be divided into the three categories machining theory-based approaches, designed experiments

Christoph Preis and Matthias Weigold contributed equally to this work.

✉ Alexander Fertig
a.fertig@ptw.tu-darmstadt.de

Matthias Weigold
m.weigold@ptw.tu-darmstadt.de

[1] Institute of Production Management, Technology and Machine Tools (PTW), Technical University of Darmstadt, Otto-Berndt-Str. 2, 64287 Darmstadt, Hessen, Germany

approaches, and artificial intelligence approaches according to Benardos and Vosniakos [4]. The artificial intelligence-based approaches consider on one side the possibility to use varying technology parameters to train models for the prediction of surface qualities. To improve the model predictions these approaches are extended by applying signals from accelerometers as model inputs [2, 14, 20, 21, 28, 37]. In addition to the prediction of the surface quality SCHUH, SCHORR, ZIEGENBEIN and BRECHER illustrate studies for predicting quality parameters, such as diameter, roundness, and concentricity of drilled and reamed holes as well as straightness of milled surfaces using internal machine tool data as input for the models' predictions [5, 6, 34, 35, 38].

However, these approaches require manual and process-specific data analysis. A high level of expertise in the area of data analysis and modeling is required for feature extraction and selection, as well as for the subsequent development of prediction models. In addition, these solutions are individually created for the respective process, which requires the manual steps to be repeated when transferring them to further processes. For this reason, methods from the field of automated machine learning (AutoML) are increasingly being used in production engineering, which enable automated prediction models to be created based on a specific input dataset. However, existing approaches usually provide domain-unspecific solutions. [24] Furthermore, to apply AutoML to time series classification, prior domain-specific feature extraction is required. The methods are designed to automatically select and train models by optimization of their hyperparameters based on a given feature-based dataset [8, 18, 22]. To overcome these issues this paper introduces an algorithm, which considers domain-specific requirements to automate the parametrization of an implemented machine learning pipeline for developing quality prediction models.

3 Machine learning pipeline

To develop the algorithm for automated pipeline parameterization, the contextualized machine tool data collected according to Fertig et al. [11] was used. A MLPL implemented with Python serves as the basis, which takes segmented time series data according to Fertig et al. [12] for each geometric element located on the workpiece along with the associated quality data. The module-based MLPL consists of submodules for feature extraction, feature selection, and machine learning, which can be parameterized via a main `PipelineConfig.json` file.

3.1 Feature extraction

In signal processing, the techniques for extracting relevant features to perform process monitoring tasks can be divided into time domain, frequency domain, and time-frequency domain methods [1, 26, 27, 36]. Typically, it is necessary to manually identify and select the appropriate features with respect to the underlying process. To reduce the manual effort and maximize the automation of the feature extraction process, the Python library TSFEL [3] is used to compute the process describing features. The extracted features are defined using a `features.json` file, in which the available features are categorized by domain into `statistical`, `temporal`, and `spectral` features.

To additionally consider information from the time-frequency domain an extension of TSFEL by the domain `temporal-spectral` is implemented. The extension applies the discrete wavelet transform (DWT) and Hilbert-Huang transform (HHT) to the input time series data. The DWT decomposes the input signal into individual frequency bands by repeating high-pass and low-pass filtering. The decomposition level determines the number of transformation steps. In each decomposition level, the high-pass filtered signal components are coded as wavelet coefficients. The low-pass filtered signal components finally serve as a basis for the subsequent decomposition step. The calculation of the wavelet coefficients $c(\tau, s)$ within the DWT is done for discrete values of the scaling parameter s and shifting parameter τ according to Eq. (1)

$$c(\tau, s) = \frac{1}{\sqrt{|s|}} \cdot \int_{-\infty}^{\infty} x(t) \cdot \psi\left(\frac{t-\tau}{s}\right) dt \qquad (1)$$

whereby $x(t)$ corresponds to the time series under investigation and ψ represents the selected wavelet basis function [10, 23, 29]. For the determination of characteristic features, the calculation of the mean, root mean square, standard deviation, kurtosis, crest factor, and the peak to peak distance for the coefficient profiles of each decomposition level is performed according to [36]. Additionally, the residual was taken into account during feature extraction.

The empirical mode decomposition as the initial step of the HHT decomposes the signal into a finite number of intrinsic mode functions (IMF) based on the sifting algorithm. To identify an IMF, the local maxima are connected via a cubic spline to create the upper envelope. Similarly, the lower envelope is obtained by using the local minima. By averaging the two envelopes, the resulting time series m_1 is obtained. The subtraction of m_1 from the original signal $x(t)$ results in the first IMF component h_1, which serves as input signal for the following iteration. These steps are repeated until a previously defined stopping criterion is reached. The second step of the HHT, is the application of the Hilbert transform to the IMFs, which uses the resulting instantaneous frequencies and instantaneous amplitudes of the signal to form the energy-frequency-time spectrum. This is known as Hilbert-Huang Transform [15–17, 33]. Due to the high output dimension of the HHT matrix, the characteristic features mean, root mean square, standard deviation, crest factor, peak to peak distance and absolute energy are calculated in the same manner as described above for the DWT [7, 31].

3.2 Feature selection

A majority of the features obtained from the automated feature extraction based on Sect. 3.1 typically yield non-relevant information regarding the prediction task or exhibit indifferent behavior under changing process conditions. Additionally, an enlarged number of input dimensions leads to an increased demand for training data and the risk of overfitting rises. Therefore, in order to achieve highest prediction performances with correspondingly high generalization capability of the predictive models, subsequent to feature extraction, the designed feature selection algorithm (illustrated in Fig. 1) is applied. It is based on the presented feature selection method from Fertig et al. [12], which has been extended within this work to ensure a more general application. Initially, all features exhibiting a variance of 0 across the analyzed dataset are removed by means of a variance threshold filter. The next step applies a set S of feature selection algorithms on the reduced feature set. S represents a subset of the implemented features selection algorithms (eg. $S = [\text{uniStat, LogisRe}]$, cf. Sect. 3.4). A final proprietary feature set (propFeatSet) is obtained for each geometric element (geomElem), located on a workpiece (cf. the workpiece shown in Fig. 3 consists of seven quality-relevant geomElem) by the set of features which are selected by the previously defined amount of $SW_{\text{fs,prop}}$ feature selection algorithms. Thereby $SW_{\text{fs,prop}}$ (eg. $SW_{\text{fs,prop}} = 2$, cf. Sect. 3.4) can be varied from 1 to the quantity of implemented and thus available feature selection algorithms.

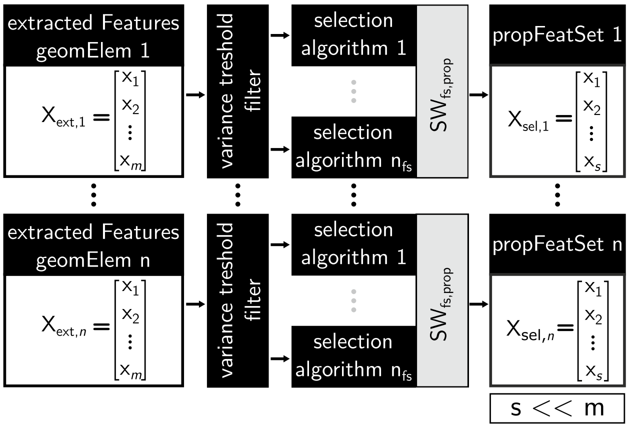
extracted Features geomElem 1
$X_{\text{ext},1} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$
variance treshold filter
selection algorithm 1
selection algorithm n_{fs}
$SW_{\text{fs,prop}}$
propFeatSet 1
$X_{\text{sel},1} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_s \end{bmatrix}$
extracted Features geomElem n
$X_{\text{ext},n} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$
variance treshold filter
selection algorithm 1
selection algorithm n_{fs}
$SW_{\text{fs,prop}}$
propFeatSet n
$X_{\text{sel},n} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_s \end{bmatrix}$
$s << m$

Fig. 1 Schematic illustration of the feature selection algorithm [12]

The implementation provides four selection algorithms, which can be combined arbitrarily. The first one consists of a univariate feature selection (uniStat) based on the determination of the mutual information between the feature vector and target variable. In addition, logistic regression (LogisRe) with elastic net regularization is utilized for feature selection. This regularization technique is particularly suited for a large number of features and a small number of training samples [39]. Lasso regression (Lasso), a shrinkade method, uses L1 and L2 regression penalty terms to shrink the coefficients of irrelevant features to 0. This model-based selection method allows a straightforward selection of the influential features by analyzing the model coefficients [13, 19]. For Lasso feature selection, the Least Angle Regression (LARS) algorithm developed by Efron et al. [9] is used to compute the coefficients, which calculates all Lasso estimates at high computational efficiency. In particular, LARS shows its advantages in high-dimensional datasets. Efron et al. [9] The fourth method for feature selection consists of the efficient wrapper approach boruta (Boruta). It aims to the identification of all relevant features for the prediction task. For this purpose, shadow features exhibiting random values are taken into account in addition to the real features. Finally, the feature selection is performed by comparing the feature importance, given by the used random forest, between the real and the shadow features. Kursa and Rudnicki [25]

3.3 ML algorithms and optimization

Adapted from Fertig et al. [12] the 6 ML algorithms, Support Vector Machine (SVM), k-Nearest Neighbors (KNN), Ridge Regression classifier (RidgeRe), Gaussian Naive Bayes classifier (GNB) Decision Tree (DT), Multilayer Perceptron (MLP) and the 3 ensemble algorithms Random Forest (RF), Extra Trees classifier (XT), AdaBoost classifier were implemented within the MLPL. The optimization of the hyperparameters for each algorithm is done using a grid search combined with stratified 3-fold cross-validation using precision as scoring function. For implementing the algorithms as well as the feature selection the python library *scikit-learn* was used [30].

3.4 Interim conclusion

By using the underlying pre-processed internal data from the machine tool and quality assurance, the developed MLPL enables automated individualized modeling for each quality-relevant geomElem on a workpiece. The following parameterization of the MLPL was applied by Fertig et al. [12]:

- feature extraction domains: `statistical`, `temporal`, `spectral` (cf. Sect. 3.1).
- feature selection : $S = \left[\text{uniStat}, \text{LogisRe}\right]$ with $SW_{\text{fs,prop}} = 2$ (cf. Sect. 3.2)
- scaling method: standardisation

The promising prediction results show the potential of the obtained models for quality prediction. Nevertheless, the numerous parameterization options of the MLPL lead to the assumption that the performance of the models can be further increased by suitable parameter combinations. After each run, 9 models are available for each geomElem. Thus the appropriate model suitable for the quality prediction must be selected to be used in the application. The presented algorithm in the following sections provides the automated parameterization of the MLPL via the `PipelineConfig.json` to obtain and select the best suitable models for the quality prediction task.

4 Algorithm for automated parametrization of MLPL

The objective of the automated parameterization is to determine the configuration that leads to the best-performing models using the MLPL for the given use case. In addition, for each geomElem, the appropriate model intended for use in data-based quality prediction will be selected, based on a domain-specifically elaborated scoring approach. This enables domain experts to efficiently create quality prediction models for individual workpieces and machine tools without additional manual intervention.

4.1 Algorithm description

The developed algorithm consists of multiple consecutive optimization steps (optSteps), which are executed sequentially to identify the optimal parameters gradually. The decision to perform a stepwise parameter optimization was motivated by the consideration that a full factorial implementation of the selected parameter combinations requires more than 600,000 MLPL runs, which leads to an unacceptable computational effort. After performing an optimization step, the corresponding identified parameters are set and the optimized configuration is applied to perform the next optimization step. Fig. 2 shows the operation procedure of the algorithm, which consists of running the MLPL, described in Sect. 3, with different configurations to identify the best-performing values for each parameter. The following six optSteps are considered within the algorithm:

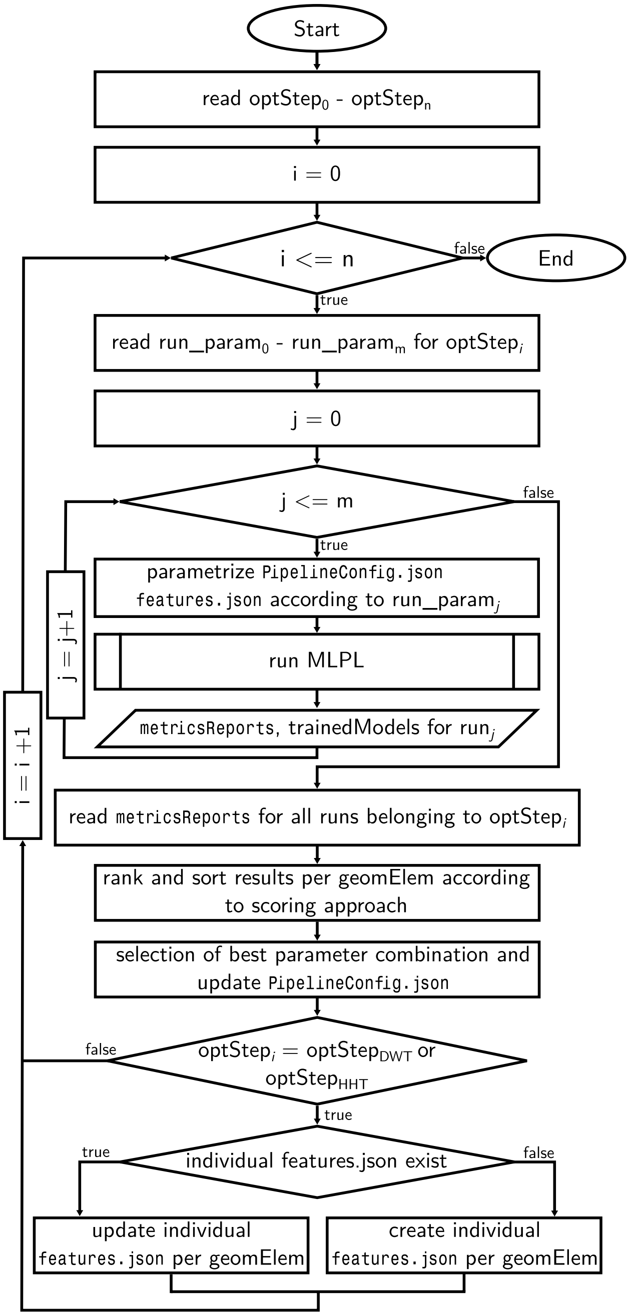
Start
read optStep$_0$ - optStep$_n$
i = 0
i <= n
false
End
true
read run_param$_0$ - run_param$_m$ for optStep$_i$
j = 0
j <= m
false
true
parametrize PipelineConfig.json
features.json according to run_param$_j$
j = j+1
run MLPL
metricsReports, trainedModels for run$_j$
i = i +1
read metricsReports for all runs belonging to optStep$_i$
rank and sort results per geomElem according to scoring approach
selection of best parameter combination and update PipelineConfig.json
false
optStep$_i$ = optStep$_{\text{DWT}}$ or optStep$_{\text{HHT}}$
true
true
individual features.json exist
false
update individual features.json per geomElem
create individual features.json per geomElem

Fig. 2 Algorithm for automated parametrization of the MLPL

- Optimization of feature extraction

  1.1 `optStep`$_{\text{DWT}}$: Optimization of DWT Hyperparameters (basis-wavelet, decomposition level)
  1.2 `optStep`$_{\text{HHT}}$: Optimization of HHT Hyperparameters (number of IMFs)

2. Optimization of domain `temporal-spectral`
3. Optimization of domains and window function to use

- Optimization of feature selection

4. Optimization of feature selection algorithms, $SW_{\mathrm{fs,prop}}$, and scaling method to use

- Final run with optimized parameter configuration of the MLPL

5. Final run and seletion of the best-performing models per geomElem $SW_{\mathrm{fs,prop}}$

These are defined in a `configTable_optSteps`. For each optStep a separate `configTable_`*`optStep`*`_runs` provides the specified parameter values `run_param` to be set for the MLPL runs. Accordingly, the MLPL is iterated within an optStep according to the number of parameter combinations of the `configTable_`*`optStep`*`_runs` with individually adjusted parameter values. The results for each run are stored as a metrics report, which contains the resulting prediction metrics. Additionally, the MLPL provides the trained models for the corresponding run. Basically, the main concept consists of a modular and extensible design, which allows the algorithm to run through additional optimization steps by extending the `configTable_optSteps` with the corresponding `configTable_`*`optStep`*`_runs`.

After performing an optimization iteration, the `metricsReports` are read for each run to rank and sort the results per geomElem following the developed scoring approach (cf. Sect. 4.2). These results are subsequently used to determine which parameter values lead to the highest performance metrics. Finally, the `PipelineConfig` gets reparameterized for the next optStep based on the identified parameters.

Both the DWT and the HHT exhibit hyperparameters, which need to be adapted to the characteristics of the available data for adequate feature extraction. For this purpose, the `optStep`$_{\mathrm{DWT}}$ thus envisages runs under altered basis-wavelets as well as decomposition levels. Included are the wavelet families daubechies, coiflet, symlet, biorthogonal with 9 shapes and the decomposition levels 3–7. Owing to the different processes resulting from the individual geometry of each geomElem, individual DWT hyperparameters are selected for each geomElem, which leads to the parameterization of the created individual `features.json` configuration files. The same applies to the HHT, in which the number of IMFs (in this case 1–7) is selected individually for each geomElem. Following the determination of the individual hyperparameters, a subsequent decision is required on whether features extracted using HHT should be considered in addition to DWT based features. Preliminary tests showed that the DWT is considerably more powerful compared to the HHT in terms of prediction performance. The combination of both time-frequency methods yielded no improvement in the results. Since it cannot be excluded that in particular cases the additional use of HHT features may achieve better performance, the described optStep 2 is included as well. The final step of optimizing the feature extraction is to select which domains should be included in the model building process. For the spectral domain, the Hanning, Hamming, and Blackman window functions are additionally examined to improve the quality of the spectral analysis [32]. The selection of domains to be considered is based on different subsets of the available domains. The subsets consist of each domain individually (4 subsets), the combinations of temporal-spectral with the other domains (3 subsets) as well as all domains together (1 subset).

The feature selection is optimized based on the feasible combinations of the four implemented algorithms. $SW_{\mathrm{fs,prop}}$ is adjusted according to the number of algorithms used per run. When using one feature selection method, a threshold $SW_{\mathrm{fs,prop}} = 1$ follows. Above two to four methods, $SW_{\mathrm{fs,prop}}$ iterates between 2 and 4 respectively. Each of the resulting 21 parameter combinations is executed once with the standardization and normalization of the scaling methods, requiring 42 runs for optimizing the feature selection. The final run finally performs the MLPL using the optimal configuration identified by the algorithm to build and select the final prediction models for each geomElem.

The modular architecture and the various configuration files as well as tables enable a flexible extension of the algorithm by simple adaptation. This ensures broad applicability by allowing additional values to be easily added to the verification procedure if desired.

4.2 Scoring and parameter value selection

At the completion of each optStep, the results need to be analyzed and the parameter values that lead to the best prediction performance need to be determined based on the `metricsReports` created for each geomElem, which contain the classification performance measures for accuracy (ACC), precision (PREC), recall (REC) and specificity (SPEC) [12]. Additionally, the ROC AUC and the number of false positive predictions (FP). The metrics within the metricsReports are determined on the validation set. To achieve the identification of the parameters, a rank is assigned to each model. The ranks are determined using the number of FP predictions, the values for specificity and accuracy, as well as ROC AUC. A lower value for the number of FP as well as higher values of the remaining metrics lead to a better rank and thus to a preferred selection as a final model of a geomElem. Due to the desired application in quality prediction, FP predictions are considered to be particularly critical in a production environment. These lead to further processing and assembly of a workpiece that has been manufactured in violation of its tolerances. As a result, it may not fulfill its function and may not be able to withstand the operating loads acting on it. For this reason, the ascending sorted ranks belonging to the number of FP forms the first basis to select the best performing models. If the number of FP is equal for several models, the subsequent sorting base considers the sum of ranks across all metrics. If this is still insufficient for unambiguous identification, the sorting procedure takes into account the ranks of ROC AUC, specificity, and accuracy for decision-making. This individual analysis for specific geomElem allows the hyperparameters of the time-frequency feature extraction methods to be determined and adjusted. The subsequent optSteps are evaluated globally across all geometric elements. Preliminary tests have shown that this individualized consideration yields significant improvements in model performance. However, no improvements and thus no advantages were obtained by individualized evaluation of the other optSteps. To reduce complexity and preserve comprehensibility, the subsequent optSteps are evaluated globally across all geometric elements. For each parameter combination, the resulting sum of FPs on the validation set predicted from the previously determined best-performing models per run is calculated across the entire workpiece thus all geometric elements. The parameter combination which leads to the lowest number of FPs is finally selected.

This domain-adapted scoring approach allows the identification of the best-performing models per geomElem within each optimization step. The underlying configuration parameters are finally set in the corresponding config files to be considered during the next optStep.

5 Results

In this chapter, the effectiveness of the presented optimization algorithm is demonstrated by applying it to different available datasets.

5.1 Datasets

The datasets were generated in the TEC-Lab of PTW using the 3-axis DMC 850 V machining center from the manufacturer DMG MORI (DMG) and the 5-axis GROB G350 2. Gen (GROB) machining center, which both are equipped with a Sinumerik 840 D control system. For data acquisition per machine, an installed edge computing solution was used, that provides the internal drive signals from the controller at a sampling frequency of 500 Hz. The respectively recorded signals and the experimental design as well as the data acquisition and matching from quality measurement can be obtained from Fertig et al. [12]. In addition to the dataset DS_{DMG1} presented in Fertig et al. [12], two further datasets were generated using the given experimental design and the considered reference geometry. DS_{DMG2} was produced approximately 1.5 years apart from DS_{DMG1} on the same machine tool. DS_{GROB} accordingly using the GROB machining center. Table 1 summarizes additional information about the datasets.

Table 1 Datasets

ID	Machine tool	Number workpieces	Production period
DS_{DMG1}	DMC 850V	200	07.2020–08.2020 (summer)
DS_{GROB}	G350	200	02.2021 (winter)
DS_{DMG2}	DMC 850V	200	03.2022 (winter/spring)
DS_{DMG3}	DMC 850V	392	09.2020–10.2020 (summer)

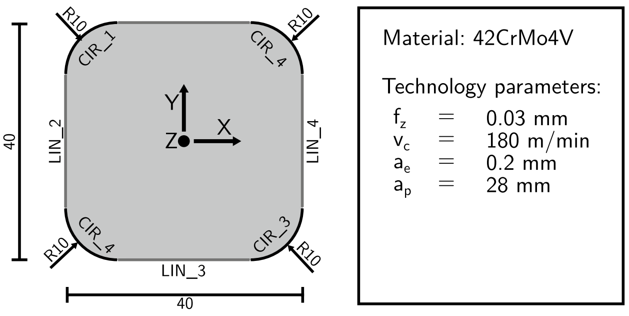

Fig. 3 Schematic representation of the produced validation workpiece “pocket”

The fourth dataset is applied for validation of the presented algorithm using a different workpiece geometry. For this purpose the pocket geometry made from the material 42CrMo4V, which is shown in Fig. 3 is considered. It consists of 7 quality-relevant geometric elements and was manufactured using the DMG machine tool. A total of 392 pockets were produced based on the previously mentioned experimental setup. For finishing the pocket, a carbide end milling tool (article number: 203089 10) made by Hoffmann Group was used. The tool features $N_z = 5$ teeth at a nominal diameter of $D_{tool} = 10$ mm. The technology parameters were summarized in Fig. 3. The pockets were arranged on 8 cuboidal plates made of 42CrMo4V with the dimensions $l \times w \times h = 305 \times 305 \times 30$ mm^3 in a 7 × 7 grid for optimized experimental procedure. This results in 392 pockets for the subsequent analysis.

5.2 Results of the automated parametrization

Figure 4 summarizes the evaluated metrics on the test dataset for each trained model obtained from the final run for each dataset. Each dot represents the corresponding score of one trained model. The black bars represent the average value across all models. The gray bar represents the results

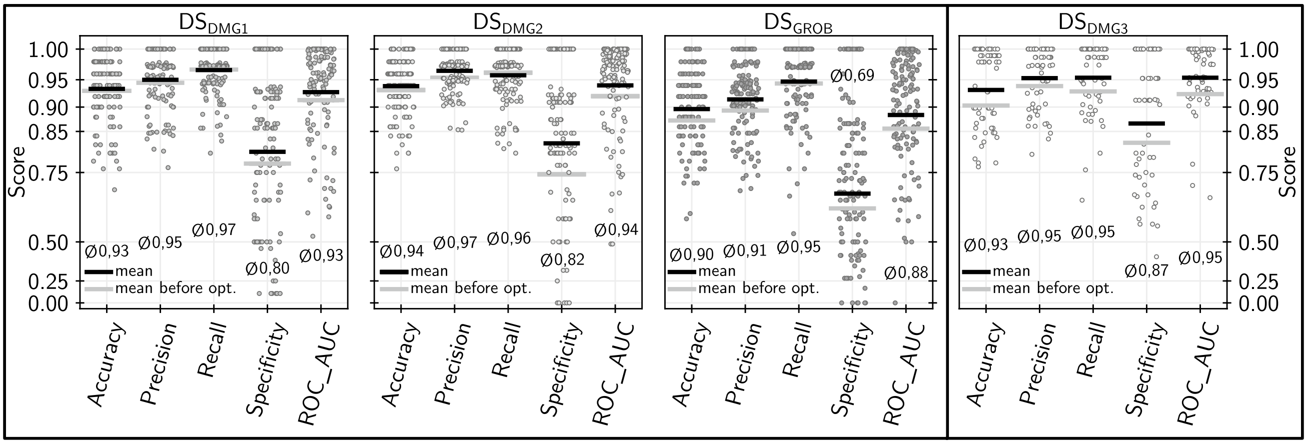

Fig. 4 Comparison of prediction scoring metrics between datasets

Table 2 Percentage improvement of metrics average by using the developed optimization algorithm

ID	Acc in%	PREC in%	REC in%	SPEC in%	ROC AUC in%
DS_{DMG1}	0.38	0.55	– 0.09	3.81	1.62
DS_{DMG2}	0.75	1.25	– 0.45	10.51	2.12
DS_{GROB}	2.63	2.40	0.35	7.74	3.30
DS_{DMG3}	3.21	1.47	2.60	5.13	3.14

of the MLPL in default configuration (cf. Sect. 3.4). The improvements in specificity and ROC AUC are clearly visible for each dataset. The average values of the other metrics also exhibit minor improvements. To analyze the improvements in more detail, Table 2 summarizes the percentage improvement of the average values relative to the default configuration. Except for the recall at DS_{DMG1} and DS_{DMG2}, the metrics in the total view across all models and geomElems show a partially significant improvement by the optimization algorithm. The improvements can be particularly observed when considering the scores of specificity and ROC AUC. This reflects the optimization target of minimizing the FP predictions since the number of FP, in this case, impacts the specificity values most. After optimizing the parameters, it is shown that for each of the 4 datasets for feature extraction, the `temporal-spectral` domain without HHT is considered solely for the prediction model development. The base-wavelets and decomposition levels selected individually for each geomElem thus provide the highest information density for quality prediction regarding the analyzed datasets. These best-performing hyperparameter values used for DWT furthermore differ among the geomElems and datasets. This supports the design of the algorithm toward individual parameter identification. In addition, the optimization algorithm selects different model algorithms for each geometric element, which leads to the conclusion that multiple model algorithms are required for the most accurate prediction.

Nevertheless, these results only provide an overall impression of the algorithm's performance. When considering Fig. 4 several cases can be identified where models show poor scores, especially for the specificity. For this reason, it is necessary to select the best performing model for each geomElem based on the presented scoring approach (cf. Sect. 4.2). Table 3 summarizes the results of the final model selection. Shown are the mean values of the achieved metrics on the test dataset across all geomElems. Additionally, the achieved metrics of the non-optimized pipeline are displayed. This clearly shows that for each data set the

Table 3 Comparison of the optimized average scores on the test dataset for the final selected models per geometric element with the non-optimized average scores

ID	ACC in% (non-opt.)	PREC in% (non-opt.)	SPEC in% (non-opt.)	ROC AUC in% (non-opt.)	FP (non-opt.)
DS_{DMG1}	97.47 (95.60)	98.77 (97.17)	94.79 (87.95)	97.32 (93.79)	0.47 (1.13)
DS_{DMG2}	98.00 (90.00)	99.66 (96.77)	98.56 (93.75)	98.85 (97.43)	0.13 (1.00)
DS_{GROB}	92.00 (78.00)	98.49 (73.33)	96.74 (68.00)	94.92 (86.40)	0.40 (2.07)
DS_{DMG3}	93.73 (77.84)	99.45 (98.58)	97.74 (96.57)	97.12 (91.24)	0.43 (0.71)

The non-optimized scores are shown in brackets optimization according to the presented algorithm leads to significant improvements regarding the final selected models. Overall, all average scores achieved high values above 90\,%. The average number of FP per geomElem is reduced to less than 0.5 for each dataset. It is worth mentioning that the FP predictions are not equally distributed among the geomElems. The total of 7 FP on the test dataset DS_{DMG1} are allocated to 5 geomElem. On DS_{DMG2}, 2 FP occurred in two geomElem and on DS_{GROB} 6 FP predictions were made on the test dataset which consists of 50 Workpieces thus 750 geomElems. The size of the test dataset was kept the same for each of these datasets. The final selected models perform on the test dataset, consisting of 98 pockets and accordingly 686 geomElems, of DS_{DMG3} only 3 FP predictions which are associated to one element.

6 Conclusion

In this paper, a new optimization algorithm for parametrization of an end-to-end machine learning pipeline (MLPL) for the development of an artificial intelligence-based quality monitoring system for milling processes is developed. The basis consists of a domain-specific developed MLPL, which automates feature engineering, feature selection and model training. The module based implementation allows to wrap the presented optimization algorithm around the MLPL. Using the preprocessed machine tool and quality data as described in Fertig et al. [11] and Fertig et al. [12] the algorithm is able to optimize the Hyperparameters of the different steps in the MLPL to train quality prediction models without the invest of manual effort by a domain expert. In particular, methods for feature engineering based on the time-frequency domain, which usually require elaborate pre-analyses of the data by experts, are parameterized and adapted to the specific task automatically using the optimization algorithm. Furthermore, the modular implementation via appropriate configuration files allows an effortless extension of the search space regarding the parameters to be optimized. The results of the analyzed four datasets demonstrate that the algorithm automatically trains and selects models with high prediction capabilities. The successful application using four different datasets further highlights the broad applicability and transferability to different machine tools and workpieces of the developed approach.

To improve the models' scores and thus deployability in production more samples of well contextualised data to increase the training datasets have to be aquired. The size of the training datasets used within this study seems to small to represent highly complex interrelationships between internal machine tool data and resulting workpiece quality in order to get even better prediction capabilities. In addition, it is necessary to examine the robustness of the models' predictions to obtain an estimation of how confident a particular model is with respect to its prediction.

Acknowledgements This research and development project ”TensorMill” is funded by the German Federal Ministry of Education and Research (BMBF) within the “Innovations for Tomorrow’s Production, Services, and Work” Program (02P17D123) and implemented by the Project Management Agency Karlsruhe (PTKA). The author is responsible for the content of this publication.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

1. Abellan-Nebot JV, Romero Subirón F (2010) A review of machining monitoring systems based on artificial intelligence process models. Int J Adv Manuf Technol 47(1–4):237–257. https://doi.org/10.1007/s00170-009-2191-8
2. Arnaiz-González Á, Fernández-Valdivielso A, Bustillo A et al (2016) Using artificial neural networks for the prediction of dimensional error on inclined surfaces manufactured by ball-end milling. Int J Adv Manuf Technol 83(5–8):847–859. https://doi.org/10.1007/s00170-015-7543-y
3. Barandas M, Folgado D, Fernandes L et al (2020) TSFEL: time series feature extraction library. SoftwareX 11(10):100456. https://doi.org/10.1016/j.softx.2020.100456
4. Benardos PG, Vosniakos GC (2003) Predicting surface roughness in machining: a review. Int J Mach Tools Manuf 43(8):833–844. https://doi.org/10.1016/S0890-6955(03)00059-2
5. Brecher C, Ochel J, Lohrmann V et al (2019) Merkmalsbasierte qualitätsprädiktion durch maschinelles lernen: Anwendung künstlicher neuronaler netze zur prozessparallelen virtuellen prüfung von qualitätsmerkmalen anhand maschineninterner daten. ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb 114(11):784–787
6. Brecher C, Ochel J, Lohrmann V et al (2020) (2020) Machinelles lernen zur prädiktion der bauteilqualität: Erweiterung eines ansatzes zur merkmalsbasierten qualitätsprädiktion durch künstliche neuronale netze. ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb 115(11):834–837
7. Dalei J, Mohanty KB (2016) Fault classification in SEIG system using Hilbert-Huang transform and least square support vector machine. Int J Electr Power Energy Syst 76:11–22. https://doi.org/10.1016/j.ijepes.2015.09.009
8. Denkena B, Dittrich MA, Lindauer M et al (2020) Using automl to optimize shape error prediction in milling processes. In: Proceedings of the machining innovations conference (MIC) 2020. https://doi.org/10.2139/ssrn.3724234

9. Efron B, Hastie T, Johnstone I et al (2004) Least angle regression. Ann Stat 32(2):407–499. https://doi.org/10.1214/009053604000000067
10. Esser C (2001) Wavelet transformation von standbildern. https://pi4.informatik.uni-mannheim.de/pi4.data/content/animations/wavelet/WaveletDemo.pdf
11. Fertig A, Kohn O, Brockhaus B, et al (2022) Consistent contextualisation of process and quality information for machining processes. In: Behrens BA, Brosius A, Drossel WG, et al (eds) Production at the leading edge of technology. Springer International Publishing, Cham, pp 195–202, https://doi.org/10.1007/978-3-030-78424-9_22
12. Fertig A, Weigold M, Chen Y (2022) Machine learning based quality prediction for milling processes using internal machine tool data. Adv Ind Manuf Eng 4:100074. https://doi.org/10.1016/j.aime.2022.100074
13. Hastie T, Tibshirani R, Friedman JH (2009) The elements of statistical learning: data mining, inference, and prediction, 2nd edn. Springer series in statistics. Springer, New York
14. Ho WH, Tsai JT, Lin BT et al (2009) Adaptive network-based fuzzy inference system for prediction of surface roughness in end milling process using hybrid taguchi-genetic learning algorithm. Expert Syst Appl 36(2):3216–3222. https://doi.org/10.1016/j.eswa.2008.01.051
15. Huang N, Shen Z, Long S et al (1998) The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. Proc R Soc Lond Ser A Math Phys Eng Sci 454:903–995. https://doi.org/10.1098/rspa.1998.0193
16. Huang NE, Attoh-Okine NO (eds) (2005) The Hilbert-Huang transform in engineering. Taylor & Francis, London
17. Huang NE, Shen SSP (eds) (2005) The Hilbert-Huang transform and its applications, Interdisciplinary mathematical sciences, vol 5. World Scientific, River Edge
18. Hutter F, Kotthoff L, Vanschoren J (eds) (2019) Automated machine learning: methods, systems, challenges. Springer eBooks Computer Science, Springer, Cham. https://doi.org/10.1007/978-3-030-05318-5
19. James G, Witten D, Hastie T, et al (2021) An introduction to statistical learning: With applications in R, 2nd edn. Springer eBook Collection, Springer US and Imprint Springer, New York https://doi.org/10.1007/978-1-0716-1418-1
20. Kant G, Sangwan KS (2015) Predictive modelling and optimization of machining parameters to minimize surface roughness using artificial neural network coupled with genetic algorithm. Procedia CIRP 31:453–458. https://doi.org/10.1016/j.procir.2015.03.043
21. Khorasani A, Yazdi MRS (2017) Development of a dynamic surface roughness monitoring system based on artificial neural networks (ANN) in milling operation. Int J Adv Manuf Technol 93(1–4):141–151. https://doi.org/10.1007/s00170-015-7922-4
22. Kißkalt D, Mayr A, Lutz B et al (2020) Streamlining the development of data-driven industrial applications by automated machine learning. Procedia CIRP 93:401–406. https://doi.org/10.1016/j.procir.2020.04.009
23. Klocke F (2018) Fertigungsverfahren 1: Zerspanung mit geometrisch bestimmter Schneide, 9th edn. VDI-Buch, Springer, Berlin http://nbn-resolving.org/urn:nbn:de:bsz:31-epflicht-1527452
24. Krauß J, Pacheco BM, Zang HM et al (2020) Automated machine learning for predictive quality in production. Procedia CIRP 93:443–448. https://doi.org/10.1016/j.procir.2020.04.039
25. Kursa MB, Rudnicki WR (2010) Feature selection with the Boruta package. J Stat Softw 36(11):1–13. https://doi.org/10.18637/jss.v036.i11
26. Lauro CH, Brandão LC, Baldo D et al (2014) Monitoring and processing signal applied in machining processes—a review. Measurement 58(10):73–86. https://doi.org/10.1016/j.measurement.2014.08.035
27. Li CJ (2006) Signal processing in manufacturing monitoring. In: Wang L, Gao RX, Pham DT (eds) Condition monitoring and control for intelligent manufacturing, Springer series in advanced manufacturing, vol 21. Springer London, London, pp 245–265, https://doi.org/10.1007/1-84628-269-1_10
28. Lo SP (2003) An adaptive-network based fuzzy inference system for prediction of workpiece surface roughness in end milling. J Mater Process Technol 142(3):665–675. https://doi.org/10.1016/S0924-0136(03)00687-3
29. Mertins A (2013) Signaltheorie: Grundlagen der Signalbeschreibung, Filterbänke, Wavelets, Zeit-Frequenz-Analyse, Parameter- und Signalschätzung, 3rd edn. Springer eBook Collection, Springer Vieweg, Wiesbaden, https://doi.org/10.1007/978-3-8348-8109-0
30. Pedregosa F, Varoquaux G, Gramfort A, et al (2012) Scikit-learn: machine learning in python https://doi.org/10.48550/ARXIV.1201.0490
31. Peng Y (2006) Empirical model decomposition based time-frequency analysis for the effective detection of tool breakage. J Manuf Sci Eng 128(1):154–166. https://doi.org/10.1115/1.1948399
32. Podder P, Khan T, Khan M et al (2014) Comparative performance analysis of hamming, Hanning and Blackman window. Int J Comput Appl 96:1–7. https://doi.org/10.5120/16891-6927
33. Quinn AJ, Lopes-dos Santos V, Dupret D et al (2021) EMD: empirical mode decomposition and Hilbert-Huang spectral analyses in python. J Open Source Softw 6(59):2977. https://doi.org/10.21105/joss.02977
34. Schorr S (2021) Prozessparallele prognose der werkstückqualität mithilfe von nc-internen daten und maschinellem lernen. https://doi.org/10.22028/D291-34543
35. Schuh G, Scholz P, Schorr S, et al (2019) Prediction of workpiece quality: An application of machine learning in manufacturing industry. In: 6th international conference on computer science, engineering and information technology (CSEIT-2019). Aircc Publishing Corporation, pp 189–202, https://doi.org/10.5121/csit.2019.91316
36. Teti R, Jemielniak K, O'Donnell G et al (2010) Advanced monitoring of machining operations. CIRP Ann 59(2):717–739. https://doi.org/10.1016/j.cirp.2010.05.010
37. Vasanth XA, Paul PS, Varadarajan AS (2020) A neural network model to predict surface roughness during turning of hardened SS410 steel. Int J Syst Assur Eng Manag 11(3):704–715. https://doi.org/10.1007/s13198-020-00986-9
38. Ziegenbein A, Fertig A, Metternich J et al (2020) Data-based process analysis in machining production: case study for quality determination in a drilling process. Procedia CIRP 93(4):1472–1477. https://doi.org/10.1016/j.procir.2020.03.063
39. Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. J R Stat Soc Ser B Stat Methodol 67(2):301–320