



In-situ identification of material batches using machine learning for machining operations

Benjamin Lutz^{1,2} · Dominik Kisskalt¹ · Andreas Mayr¹ · Daniel Regulin² · Matteo Pantano² · Jörg Franke¹

Received: 28 February 2020 / Accepted: 19 November 2020 / Published online: 26 December 2020
© The Author(s) 2020

Abstract

In subtractive manufacturing, differences in machinability among batches of the same material can be observed. Ignoring these deviations can potentially reduce product quality and increase manufacturing costs. To consider the influence of the material batch in process optimization models, the batch needs to be efficiently identified. Thus, a smart service is proposed for in-situ material batch identification. This service is driven by a supervised machine learning model, which analyzes the signals of the machine's control, especially torque data, for batch classification. The proposed approach is validated by cutting experiments with five different batches of the same specified material at various cutting conditions. Using this data, multiple classification models are trained and optimized. It is shown that the investigated batches can be correctly identified with close to 90% prediction accuracy using machine learning. Out of all the investigated algorithms, the best results are achieved using a Support Vector Machine with 89.0% prediction accuracy for individual batches and 98.9% while combining batches of similar machinability.

Keywords Process monitoring · Machine learning · Material batches · Machining

Introduction

In metalworking, the material properties of different batches might vary with significant impact on the respective metalworking process. These observations can be explained by deviations in the material's manufacturing procedure among different suppliers as well as among batches from batch production at a single supplier. In the material's manufacturing process, various factors, such as the chemical composition, the fabrication procedure, or the heat treatment might deviate slightly within their tolerances. These effects lead to small changes of the material's properties, such as microstructure, grain size, and hardness, which directly impact a material's machinability (Schneider 2002).

In their study, Goppold et al. (2018) investigate batches of metal sheets from multiple vendors, specified as the same material, finding deviations in their chemical compositions.

These deviations strongly influence the laser cutting process but can be compensated by batch-specific adaption of cutting parameters (Goppold et al. 2018). Similarly, it is found that during the hardening process of gear pinions, small fluctuations of the copper content within the material's tolerance among different batches significantly influence their hardenability (Šuchmann and Martinek 2014).

In subtractive manufacturing processes, different process behaviors among batches of the same specified material can be observed as well (Jemielniak and Kosmol 1995). While one batch of the raw material might be easy to machine with a given set of cutting parameters, a different batch might show unstable machining, increased tool wear, or even tool breakage. Thus, when optimizing a subtractive manufacturing process with regard to the machinability of one material batch, non-ideal behavior can be expected when machining a batch with different machinability, using the same parameters found before. However, as the material deviations resulting in these differences in machinability are within the given tolerance of the specified material, they cannot be distinguished without further investigation. Thus, without additional knowledge, each produced material batch from each supplier needs to be considered as a unique material batch with potentially different machinability.

✉ Benjamin Lutz
benjamin.lutz@faps.fau.de

¹ Institute for Factory Automation and Production Systems, Friedrich-Alexander-University Erlangen-Nürnberg, Egerlandstraße 7-9, 91058 Erlangen, Germany

² Siemens AG, Otto-Hahn-Ring 6, 81739 Munich, Germany

To handle these uncertainties, in practice typically one of the following approaches is carried out: either the assigned tolerances for the specified material are tightened, a set of sub-optimal parameters is used for all batches, or material testing is carried out for every single batch. While tightening the material tolerances might reduce the magnitude of batch deviations, the manufacturing costs and testing efforts increase on the material supplier's side (Dong et al. 1994). Thus, looser tolerances decrease the costs and reduce the testing efforts but increase the effect of material batch deviations. Assuming that all material batches behave similarly enough, one set of sub-optimal parameters can be determined and used for all batches. This way, the effort and costs for material characterization are minimized, however, the overall manufacturing costs might increase due to additional maintenance expenses and process operation under non-ideal conditions, also worsening the product quality. Finally, while carrying out material testing for every batch individually enables the determination of batch-specific optimized cutting parameters, the resources needed for material characterization increase. Furthermore, even material batches with the same machinability as previous batches have to be tested, as no prior knowledge about their machinability is available.

Ideally, previously found ideal parameters are reused for future batches of similar machinability. Thereby, material characterization and determination of the respective ideal cutting parameters needs to be carried out only once for batches of similar machinability. This way, every material batch can be machined with optimized parameters while reducing the experimental efforts to a minimum. To achieve this, it is necessary to identify a given work material during operation by comparing it to previously machined batches through process observation.

The machinability of a material can be assessed through the tool-lifetime, the cutting forces, the surface finish, and the chip form (Schneider 2002). Typically, tool-lifetime experiments, as described in the ISO Standard 3685, are performed. A simple cut with constant cutting conditions is carried out with a new tool, measuring the machining time needed until the tool reaches its defined end-of-life criteria. Repeating this procedure for various cutting speeds, multiple data points of cutting speed and the respective machining time until failure are recorded. This data is used to fit a process model, such as the Taylor model (Taylor 1906).

To identify changes in the machinability during machining, a process monitoring system is needed. In their review, Teti et al. (2010) cluster existing approaches based on the monitoring scope. Most publications that were found deal with tool conditions, chip conditions, process conditions, surface integrity, machine tool state, and chatter detection. Only a few approaches exist for the monitoring of the work material.

For the monitoring of different materials, Kramer (2007) investigates acoustic emission signals while machining compound parts containing steel and ceramic regions. Characteristic dominant frequencies can be found for each material, enabling the correct identification of both materials. Similarly, visual properties of the work material can be observed using machine vision and analyzed by machine learning (ML) to classify the material. In their study, Penumuru et al. (2020) evaluate such an approach using Support Vector Machines to distinguish between aluminum, copper, mild steel rusted, and medium density-fiberboard. The detection of the two metals steel and aluminum is also possible by monitoring and evaluating the cutting force, spindle torque, and material removal rate (Denkena et al. 2018). The latter approach can be further improved by the integration of accelerometers while using ML for data analysis (Denkena et al. 2019).

Focusing on different alloys of the same material, Denkena et al. (2019) monitor the vibration of the tool, process forces, and control signals, such as spindle torque and motor currents, to identify two different steel alloys. Therefore, various types of features and ML algorithms are investigated. They find the differences in material properties to be much smaller for the alloys of the same material, compared to the two different materials steel and aluminum. Thus, the process forces are rather similar for the two alloys, resulting in lower prediction accuracy. Only the investigated k-Nearest-Neighbors algorithm shows promising results for online identification. (Denkena et al. 2019).

Looking at the level of material properties, Teti and La Commare (1992) analyze a fused signal of acoustic emissions and cutting forces. Using a k-Means Clustering algorithm, samples of different heat treatments can be correctly identified. Kothuru et al. (2018) investigate the monitoring of audible sound signals for detecting hardness variations of the work material. With the investigated ML models, such as Support Vector Machines and Convolutional Neural Networks, four different hardness levels within a workpiece can be detected (Kothuru et al. 2018).

In summary, it can be said that there are some approaches in research for differentiating among material types, recognizing different alloys of the same material type, and identifying certain material properties directly. However, no research has been found for the identification of material batches of the same specified material in subtractive manufacturing. The existing approaches in industrial settings regarding this issue lead to increased manufacturing costs either due to ignorance of batch deviations, tight tolerances, or due to labor-intense experiments for repetitive material batch characterizations.

Thus, a smart service is proposed in this paper for the in situ classification of batches of the same material. This service integrates well into the concept of smart machine

tool systems, by monitoring and analyzing data while generating knowledge for improving the manufacturing process (Jeon et al. 2020). As modern machine tools already provide a variety of internal signals, such as current or torque values of linear and rotary axes (Tong et al. 2020), these signals are chosen and analyzed, as thereby the integration of this service into existing systems is reduced to software components. For the data analysis, ML methods are investigated. ML has become a common tool for analyzing industrial data, with many existing use cases in production and especially machining (Mayr et al. 2019). In machining, ML is used for applications such as predictive maintenance, quality prediction, process monitoring, or parameter optimization (Kibkalt et al. 2018; Yao et al. 2019).

In Sect. 2, the architecture of the proposed smart service is detailed. In Sect. 3, the methodology used for evaluating the proposed concept of using supervised learning for solving the issue of batch identification and classification is explained. The results are presented in Sect. 4 and discussed in Sect. 5. Finally, Sect. 6 concludes the presented research and outlines future research activities.

Service architecture

In this study, a smart service is proposed for in situ classification of the material batch during machining. An overview of the concept can be seen in Fig. 1. The proposed architecture consists of three main parts: the *data acquisition* (Fig. 1a), the *data analysis and batch identification* (Fig. 1b), and an interface to the human operator, the *human–machine interface* (HMI) (Fig. 1c).

The machine tool's internal sensors are used as the data source in the data acquisition module where the machine's *numeric control* (NC) is being used as an interface (Fig. 1a). From the NC, data about the current cutting conditions (e.g. cutting speed and feed rate) as well as process data (e.g. position and torque for each linear and rotary axis of the machine tool) can be observed.

The data is acquired continuously by the NC and forwarded to the connected edge device. There, the *data handling* service reads the incoming data packages and temporarily stores the data in the *buffer*. Once data is available for a predefined period of time, the respective data within that window is forwarded to and analyzed with the *data analysis and batch identification* routine.

In the *data analysis and batch identification* routine (Fig. 1b), the process data is preprocessed and analyzed, and the batch is identified. In the *data aggregation* module, a rolling mean is applied on the torque time-series for each machining axes to create features. These feature vectors are now preprocessed in the *data preprocessing* module. This involves scaling the feature vectors with a pre-fitted *stan-*

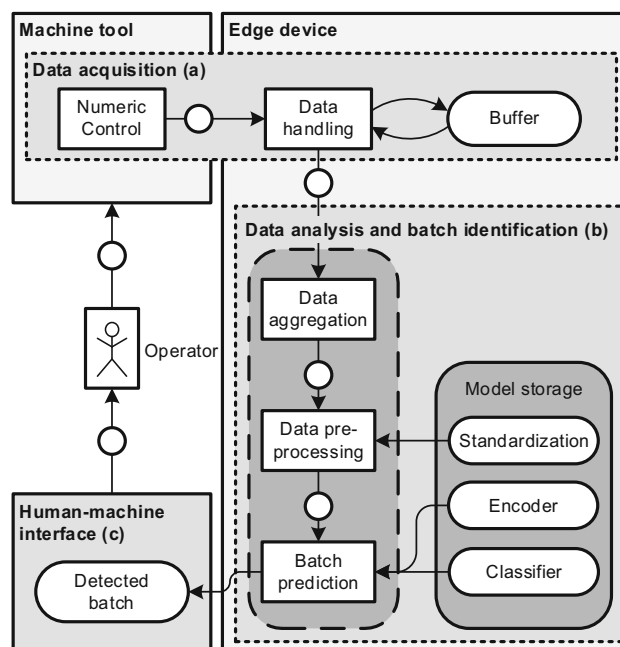


Fig. 1 Architecture of the smart service for in situ identification of the material batch displayed as block diagram according to the Fundamental Modeling Concept (Knöpfel et al. 2005)

ardization to improve the prediction accuracy. By scaling, the imbalance of features showing large differences with features showing only small differences can be compensated. Lastly, the preprocessed features are used to identify the material batch in the *batch prediction* module. This module uses a pretrained *classifier* to classify the batch. Using the fitted *decoder*, the prediction result is decoded to the detected batch. The respective models for data preprocessing and classification have to be fitted prior to operational use and are stored in the *model storage*.

Finally, the prediction about the material batch currently machined is conveyed to the machine's operator through an *HMI* (Fig. 1c). The operator can now use this information to adapt the cutting process according to the known characteristics of the detected material batch.

Methodology

Consecutively, the data analysis and batch identification module as a key component of the proposed concept for in-situ identification of material batches is implemented and evaluated. The used procedure can be seen in Fig. 2.

Cutting experiments

Experiments are conducted using a computerized numeric control (CNC) lathe. The lathe is equipped with a cutting

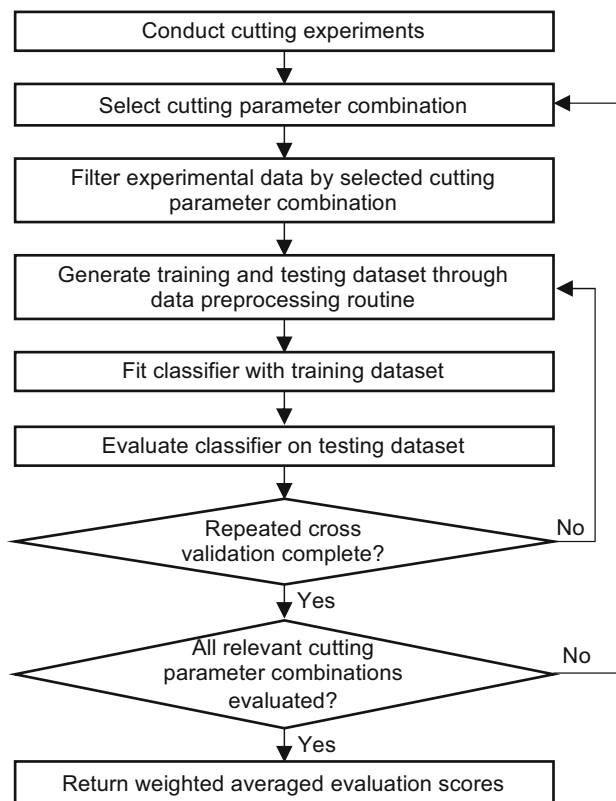


Fig. 2 Training and evaluation procedure investigating each POI on its own with cross-validation carried out on experiment level

tool using exchangeable inserts. During machining, the process data, containing information about the torque of all machining axes, is acquired with a frequency of 500 Hz. Additionally, the commands issued to the machine, especially the unplanned interactions of the machine's operator, are stored. After each experiment, the condition of the cutting tool insert, expressed as flank wear width, is measured with the procedure described in Lutz et al. (2019). A unique identifier, batch A–E, is assigned to each investigated material batch.

The experiments are carried out as sets of experiments. Each set is started with a fresh insert and stopped at varying degrees of tool wear. While the majority of experiments is carried out until the insert reaches its end-of-life criteria, others are stopped early, thus the data density is increased as more combinations of cutting conditions are investigated. Within a set of experiments, the cutting conditions, type of cutting tool insert, cutting depth, cutting speed, feed rate, and material batch, are kept constant. Only the tool condition, expressed as flank wear width, worsens from start to end of each set. Among all sets of experiments, combinations of two different insert types (type A and type B), five material batches (batches A–E), one cutting depth (4 mm), ten feed rates (from 0.1 to 1.0 mm/rev in 0.1 mm/rev increments), and

eight cutting speeds (from 180 to 300 mm/min in 20 mm/min increments as well as 350 mm/min) are investigated.

Due to resource constraints, not all combinations of cutting parameters and material batches are explored. Seven combinations of cutting conditions are studied with a high volume of experiments as seen in Table 1. 16 additional combinations of cutting parameters are investigated with fewer experiments. As previously described, for these cutting parameter combinations only incomplete sets of experiments are carried out. For each combination of cutting conditions investigated in this manner, between two and six experiments for three to five material batches are carried out.

Data preprocessing

As described in Sect. 2, the data needs to be adjusted and preprocessed. First, the preset cutting speeds and feed rates are corrected into the actual values taking the machine operator's override function into account. The flank wear width is interpolated linearly within each experiment. For the torque signals, redundant information is discarded, as some machining axes contain multiple motors with similar values.

Once all the signals are adjusted, the data preprocessing is carried out following the procedure in Fig. 3. To generate the training and testing datasets for a given combination of cutting conditions (metadata), all available data from the experiments is filtered by the given metadata. Thus, only experiments conducted at similar cutting conditions are used for model training and testing. All selected experiments are now split into experiments that will be used for model training, and experiments that will be used for model testing. As the separation in training and testing dataset takes place at experiment level with cuts being conducted at different parts of the workpiece and not on the sample level with samples from similar positions, over fitted models perform worse on the testing dataset and can thus be discarded during hyperparameter optimization.

Consecutively, a sliding window approach is used to create discrete samples from the time-series signal. Here, windows with a length of 400 ms, containing 200 datapoints, are used. All signals are aggregated by calculating the respective mean values within the selected window as features. The aggregated torque values from the machining axes contain the relevant information about the cutting process that will be used to determine the material batch.

As the material batch identifier is a categorical value that will be used as a label for the supervised learning approach, it is encoded using a one-hot encoder (Witten et al. 2011). During the training phase, the characteristic vectors are scaled by standardization to a standard distribution with a mean of zero and a variance of one by subtracting the mean of all samples μ from each value x_i and dividing it by the standard deviation σ (1). Thereby, the training process can be improved (Witten

Table 1 Overview of the conducted experiments at cutting parameter combinations with a high volume of experiments

Insert type	Cutting speed (m/min)	Feed rate (mm/rev)	Wear	Batches	Experiments	Samples
A	180	0.7	200 μm ± 100 μm	A, B, C	70	7161
A	240	0.7	200 μm ± 100 μm	A, B, and D	20	2060
B	240	0.5	200 μm ± 100 μm	C, D, and E	86	7822
B	240	0.7	150 μm ± 100 μm	A, B, C	29	1986
B	300	0.5	250 μm ± 100 μm	B, C, and E	96	3146
B	300	0.7	50 μm ± 100 μm	A, B, C, D, and E	33	1010
B	350	0.5	150 μm ± 100 μm	B, C, D, and E	42	1181

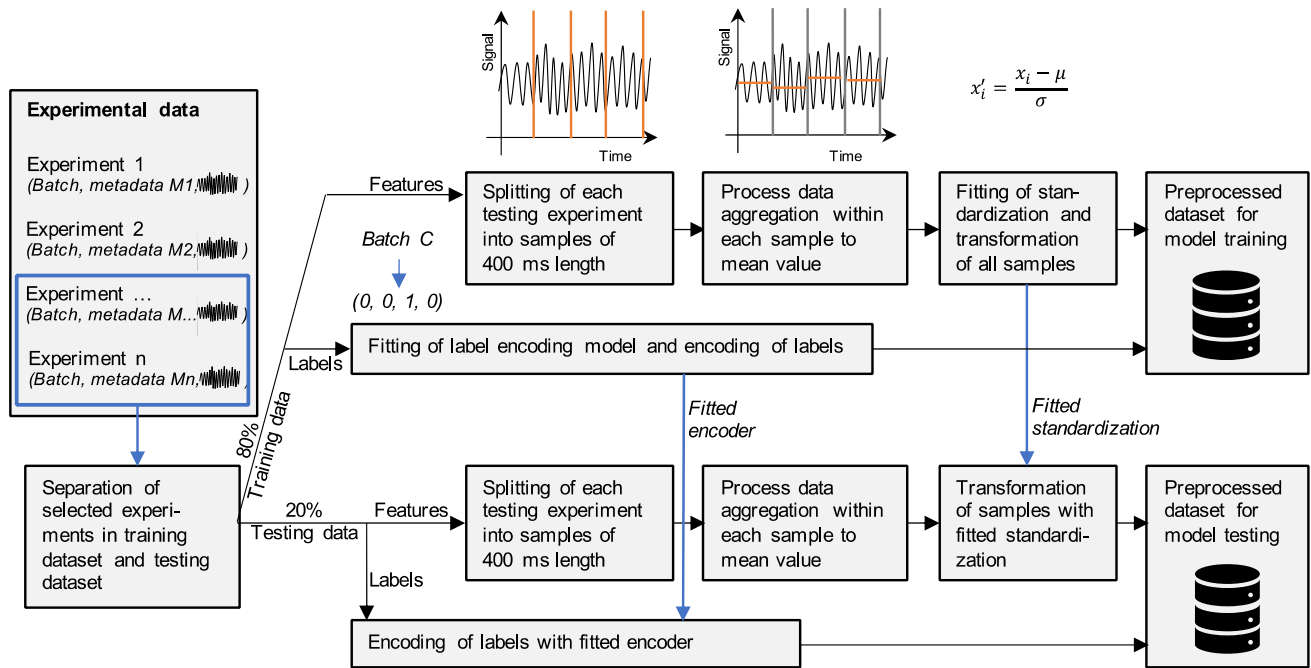


Fig. 3 Detailed overview of the data preprocessing procedure

et al. 2011). Both the encoder and the scaler are only fitted on the samples of the training dataset but used to transform both the training and the testing dataset. Finally, all preprocessed samples contain the torque values, the metadata, and the one-hot encoded material batch identifier.

$$x'_i = \frac{x_i - \mu}{\sigma} \tag{1}$$

Model definition

In the present approach, supervised learning models are investigated for identifying the material batch. The models are trained for a single combination of cutting conditions, using the metadata to filter relevant experiments for generating the training and testing dataset (see Sect. 3.2). Thus, the metadata are not used as an input for the model but rather to filter relevant training data. Only the preprocessed torque

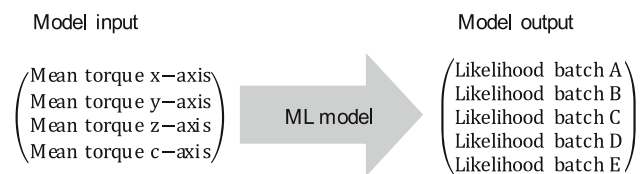


Fig. 4 General design of the classification model

time-series values are considered as input features for the models, while the one-hot encoded material batch identifiers are the targets to be predicted by the classifiers (Fig. 4).

In terms of models, established ML models are investigated and compared to a simple Logistic Regression (LR) model as a baseline. The investigated models include Support Vector Machines (SVM), Random Forests (RF), Naïve Bayes Classifiers (NB), k-Nearest-Neighbors (kNN), Artificial Neural Networks (ANN), and Radial Basis Function

Neural Networks (RBFNN). The RBFNN network consists of a single hidden layer using k-means for center initialization. The designed ANN consists of multiple hidden layers, with each consecutive hidden layer having twice as many neurons as the previous one. For the input layer and all hidden layers, the ReLU activation function is used, while for the output layer the softmax activation function is applied. Dropout is used to prevent overfitting (Srivastava et al. 2014). After initialization, each model is optimized regarding its respective hyperparameters.

The models are compared using a repeated k-fold cross-validation or repeated leave-one-out cross-validation strategy, depending on the amount of training data available at the selected cutting conditions. For 20 experiments and more, a ten-fold cross-validation strategy is used. Otherwise, a leave-one-out validation strategy is used due to the limited number of experiments. Both validation strategies are repeated ten times to compensate for statistical effects in data splitting and model training. Once the validation procedure is completed for every combination of cutting conditions, the weighted averaged evaluation scores are computed. Here, the sample quantity of the respective combination of cutting conditions is used as weight. Besides calculating the prediction accuracy, the computation time needed to train the model is also investigated. The training times for cutting parameter combinations with high and low data volume are reported separately, as the differences in volume might influence the computational effort needed for creating the models. Additionally, the inference times for each model are recorded. All models are trained on the same hardware.

Results

Machinability investigation

First, the material batches are investigated regarding their differences in machinability. Within the sets of experiments where the cutting tool inserts were used from fresh to worn-out, the total lifetime of each insert is measured. In Fig. 5, the Taylor curves are displayed for a feed rate of 0.7 mm/rev (Fig. 5a) and a feed rate of 0.5 mm/rev (Fig. 5b) for different types of cutting tool inserts. Besides batch A, all batches align well with the fitted curve.

As assumed, large differences in machinability can be observed. For a feed rate of 0.7 mm/rev and a cutting speed of 300 m/min using insert type A, a similar trend among batches B and C can be observed. The machinability of batch B, having five minutes lifetime, is higher than batch C with a lifetime of 3 min. Batch A also shows good machinability, however, for lower cutting speeds such as 180 m/min, batch B lasts 66 min, whereas batch A only lasts 33 min. For a feed rate of 0.5 mm/rev and a cutting speed of 300 m/min, batch B

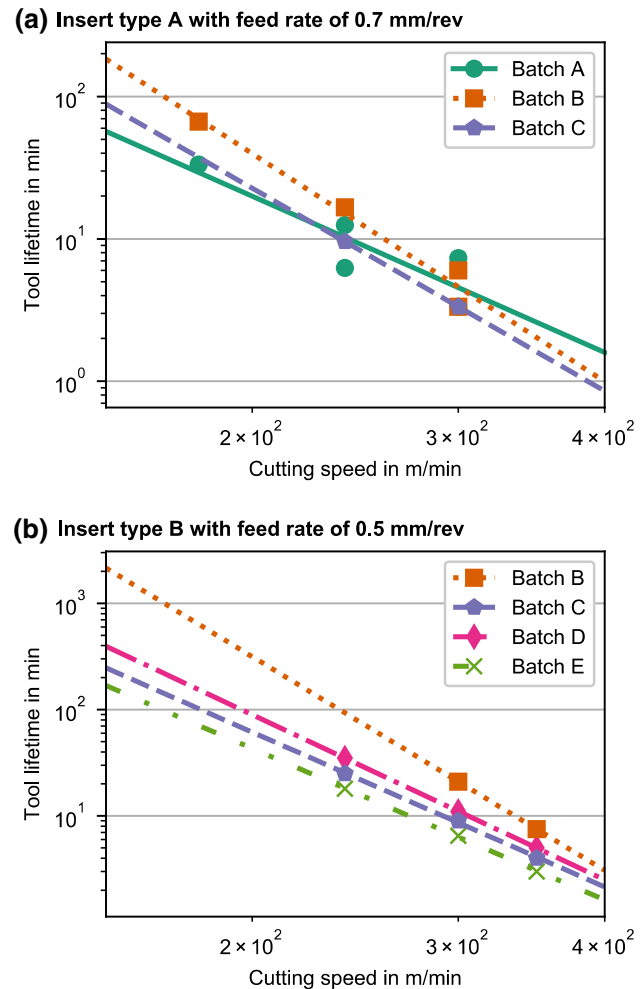


Fig. 5 Calculated machinability for the tested material batches showing differences in insert lifetime for similar cutting speeds

shows the highest machinability with 21 min of tool life. In comparison, batches C, D, and E only last 9, 11, and 6.5 min, respectively.

Additionally, in Fig. 5b it can be observed that batches C, D, and E show rather similar behavior compared to batch B. For a cutting speed of 300 m/min batches C, D, and E exhibit an insert lifetime of around 9 min \pm 2 min compared to 21 min for batch B. For a cutting speed of 350 m/min, batches C, D, and E exhibit an insert lifetime of around 4 min \pm 1 min compared to 7.5 min for batch B.

Model optimization

For building the material identification model, ML techniques are used. The algorithms are implemented using Python with the *scikit-learn* library (Pedregosa et al. 2011). The ANN is implemented using *Tensorflow* (Abadi et al. 2016) and *Keras* (Chollet 2015). Each algorithm is tuned by investigating multiple hyperparameters in a grid search

Table 2 Investigated and optimal hyper parameters for each model

Algorithm	Parameter	Investigated parameters	Optimal parameter
SVM	C	1e3, 5e2, 1e2, 5e1, 1e1, 1, 1e-1, 1e-2	5e2
	Kernel	Linear, Poly, RBF	Poly
	Degree	2, 3, 4, 5, 7	2
	Gamma	Scale, 1e2, 1e1, 1, 1e-1, 1e-2	Scale
RF	Estimators	10, 50, 100, 150, 200, 500	150
	Criterion	Gini, entropy	Gini
	Max depth	4, 5, 7, 9	5
	Max features	2, 3, 4, 5	5
NB	Var smoothing	1e-3, 1e-4, 1e-5	1e-4
kNN	Neighbors	1, 5, 14, 39	39
	Weights	Uniform, distance	Distance
	Leafsize	15, 30, 50	15
	P	1, 2, 4	1
	Metric	Minkowski, cosine	Minkowski
ANN	Neurons	16, 32, 64, 128, 256, 512	256
	Hidden layers	1, 2, 4	2
	Dropout	0.0, 0.2, 0.4	0.4
	Optimizer	RMSprop, Adam, SGD	Adam
RBFNN	Neurons	4, 8, 16, 32, 64, 128, 256, 512	32
	Beta	0.01, 0.05, 0.1, 0.5, 1.0, 2.0, 5.0	2.0
	Initializer	Random, k-means	k-means

manner and rated using the previously introduced cross-validation strategies. All investigated hyperparameters and the ones with the best results are reported in Table 2 for each algorithm.

Besides optimizing the hyperparameters of the ML models, the proposed system can be adapted based on the criteria for selecting experiments for model training. Experiments with exactly the same cutting speed, feed rate, and type of cutting tool are selected for a given combination of cutting conditions by default. Only for the tool condition, experiments within a range of $\pm 100 \mu\text{m}$ of the defined flank wear width are included. However, besides considering only experiments with exactly matching cutting speed and feed rate, tolerances can be applied to those as well to increase the number of training data. The effect of widening the toler-

ances for the cutting speed, the feed rate, and the flank wear width is shown in Fig. 6. The results are reported separately for combinations of cutting conditions with a high data volume (Fig. 6a–c) and a low data volume (Fig. 6d–f).

It appears that for the cutting conditions with high data volume, most algorithms do not benefit from a widened tolerance of the cutting speed but rather show the best results with exactly matching speeds at a tolerance of $\pm 0 \text{ m/min}$ (Fig. 6a). Besides the LR, all algorithms perform significantly worse for the broadest tolerance of $\pm 60 \text{ m/min}$. For the feed rate (Fig. 6b), similar results can be observed. In contrast, for the flank wear width, a different behavior can be seen (Fig. 6c). The SVM, ANN, RF, and kNN algorithms all show the best results at the middle default tolerance level of $\pm 100 \mu\text{m}$, with rather similar prediction results across all tolerances. The NB algorithm shows similar, good results for $\pm 50 \mu\text{m}$ and $\pm 175 \mu\text{m}$ but a worse performance for $\pm 100 \mu\text{m}$. The RBFNN shows decreasing prediction performance with broadening the tolerance, while the LR shows the opposite trend.

For cutting conditions with low data volume, all algorithms seem to greatly benefit from broader tolerances. For the cutting speed (Fig. 6d), an increase in accuracy between 15 percentage points (p.p.) and 25 pp. can be observed by increasing the tolerance from ± 0 to $\pm 60 \text{ m/min}$. For all algorithms, the best results are achieved with tolerances of $\pm 60 \text{ m/min}$, considering nearly all available data. For the feed rate (Fig. 6e), the strongest prediction improvement with broader tolerances can be seen. The SVM, ANN, and LR show improvements above 40 p.p., while the remaining algorithms show improvements between 20 and 30 pp. for increasing the tolerance level from ± 0.0 to $\pm 0.2 \text{ mm}$. Lastly, the flank wear width (Fig. 6f) seems to significantly influence the SVM, ANN, RF, kNN, and LR, where an increase of 10 pp. can be seen between ± 100 and $\pm 175 \mu\text{m}$. For the other algorithms, only slight increases can be observed. Between the tolerance levels of ± 50 and $\pm 100 \mu\text{m}$, no significant differences can be detected.

Comparing the accuracies among the models, similar trends can be seen. For the cutting conditions with high data volume, the best results are achieved by the SVM, followed by the ANN and the RF with rather similar prediction accuracies. The remaining algorithms kNN, NB, and RBFNN show similar or worse performance compared to the LR. For the cutting conditions with low data volume, most algorithms can't differentiate among the different batches. Only for the highest level of tolerances prediction accuracies above 60% can be seen for the SVM, ANN and LR.

Summarizing the findings, the SVM performs best for large amounts of data, which can be seen by the high accuracies for the cutting conditions with high data volume and the steep increase in accuracy for low data volume cutting condi-

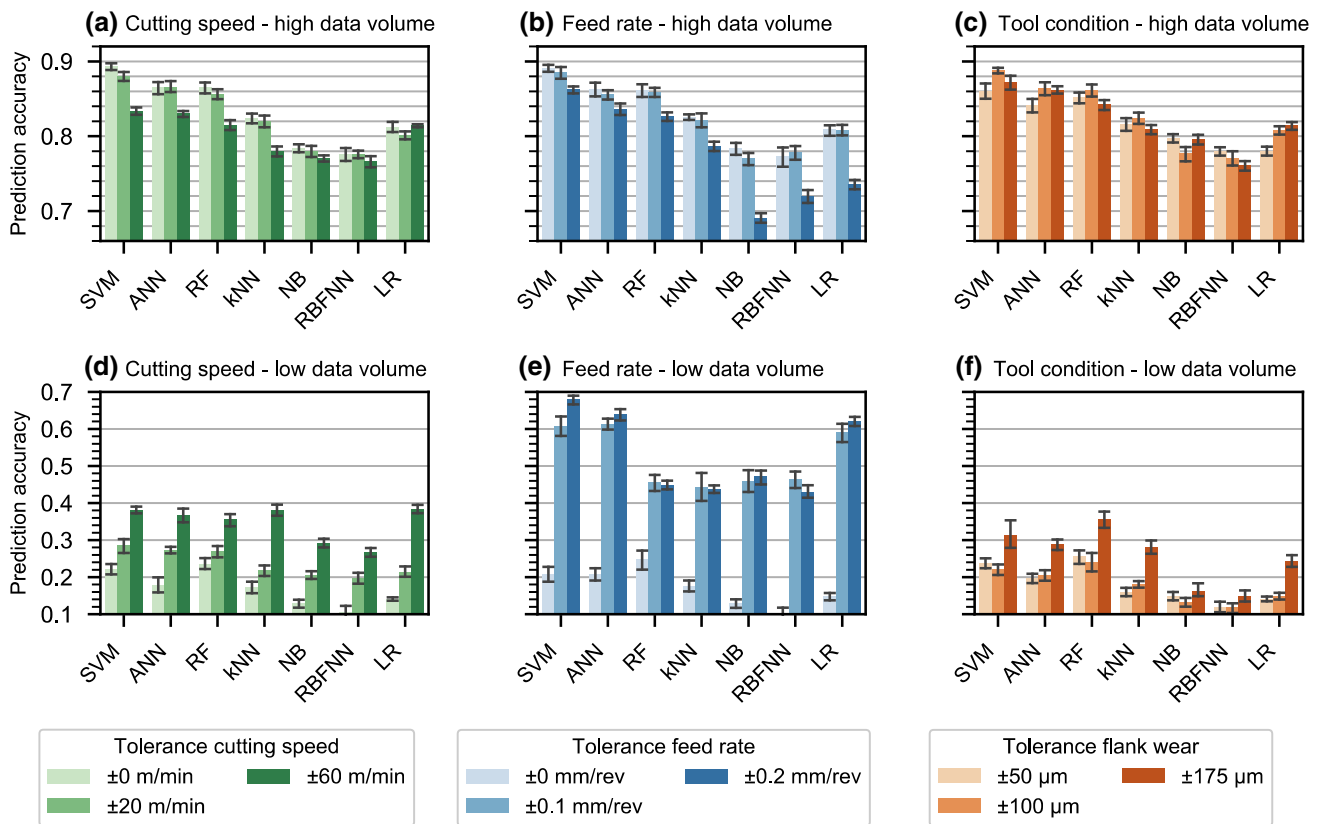


Fig. 6 Influence of increasing the tolerance for including data in the model training on the resulting prediction accuracy for cutting condition combinations with high data volume (a–c) and low data volume (d–f)

tions in combination with high tolerances, thus considering the most data for training.

Evaluation of prediction performance

After optimizing the hyperparameters and tolerances for each algorithm, the final metrics for evaluation are computed (Table 3). The best results can be achieved using the SVM, producing an accuracy of 88.9% averaged over all combinations of cutting conditions with high data volume, followed closely by the RF with 86.1% and the ANN with 85.7%. Thereby, these models outperform the LR as a baseline with a prediction accuracy of 81.7% by between 4 and 8 pp. For the remaining cutting conditions with low data volume, the best results are achieved with the ANN with 71.0%, closely followed by the SVM with 70.3% and the LR with 68.9%.

As the best model, the SVM is used to compute the confusion matrix (see Fig. 7). The confusion matrix displays the predicted batch identifiers versus the true identifiers. Here, differences among the batches can be observed. Batch A can be identified nearly perfectly with an accuracy of 99.9%. Similarly, batch B shows a correct identification rate of 97.3%. Batch C, D, and E show prediction accuracies of 86.0%, 80.0%, and 73.8%, respectively. Analyzing the errors

of these material batches, the incorrect predictions are not split evenly among the remaining batches but are concentrated on a few. For batch E with an accuracy of 73.8%, 6.9% of all samples are classified as batch C and 18.6% as batch D with close to no predictions for batch A and B.

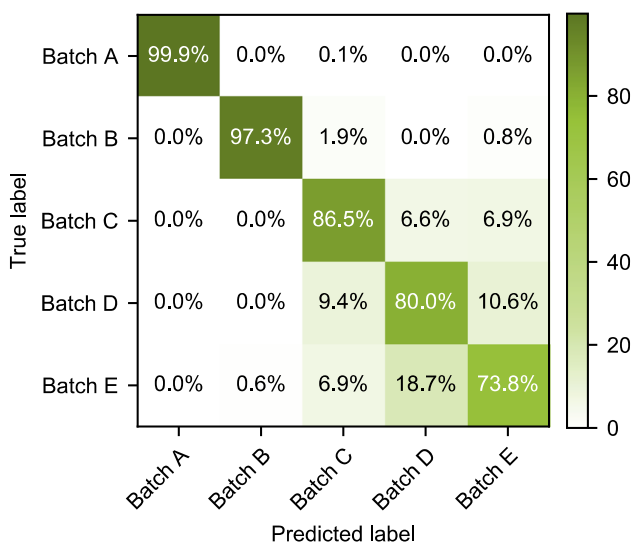
Evaluation of computational performance

For operational use, the computation time for model training and inference are compared (see Table 3). The times for low data volume represent a newly set-up system with little data available, whereas the times for high data volume represent a version with a large experience database. The average training dataset contains 2785 samples for high data volume and 1585 samples for low data volume after filtering.

For training the models with the high data volume, two groups can be observed. NB and kNN show the fastest training with 6 ms and 10 ms, respectively. The remaining algorithms are up to three magnitudes slower in training with 0.4 s for the LR, 2.6 s for the RF, 4.6 s for the RBFNN, 5.3 s for the SVM, and 20.7 s for the ANN. For a low data volume, similar results can be observed among the algorithms. Comparing the training times between high and low data volume, no significant differences are seen for the RF, NB, kNN,

Table 3 Comparison of the different models regarding their accuracy, training time, and inference time for high data volume (vol.) and low data volume cutting conditions

Algorithms	Accuracy		Training time		Inference time	
	High data vol.%	Low data vol.%	High data vol. (s)	Low data vol. (s)	High data vol. (ms)	Low data vol. (ms)
Support vector machine	89.0	70.3	5.2	1.8	0.7	0.9
Random forest	86.1	62.8	2.6	2.3	52.9	56.8
Naive Bayes	81.2	47.3	0.6×10^{-2}	0.6×10^{-2}	0.7	0.8
K-nearest-neighbors	82.6	63.6	0.1×10^{-1}	0.8×10^{-2}	2.3	2.7
Artificial neural networks	85.7	71.0	20.7	10.8	223.0	199.6
Radial basis function neural networks	76.7	58.0	4.6	4.7	234.0	270.3
Logistic regression	81.7	68.9	0.4	0.4	0.4	0.4

**Fig. 7** Confusion matrix showing the prediction accuracy of the batch identification algorithm

RBFNN, and LR, while the training times for the SVM and the ANN are reduced by a factor of 3 and 2, respectively.

For inference, as expected, no significant differences between high and low data volume can be observed. Comparing the different algorithms, the SVM, the kNN, the NB, and the LR all have inference times of about a millisecond. The inference times for the RF, ANN, and RBFNN are between one and two magnitudes higher.

Discussion

Looking at the tolerances used for filtering the historic training database, different behavior depending on the data volume for the selected cutting conditions can be seen. It is shown that for cutting conditions with a natively high data

volume, lower tolerances achieve higher accuracies. Thus, as expected, data that is the most similar to the cutting conditions under investigation provides the most information about the machined batch, as there is no influence on the signal from changed machining conditions.

For natively low amounts of training data available, such as cutting conditions that are seldomly used, or during the set-up phase of the system with little historic data available, low prediction performances are observed, especially with tight tolerances. Even simple algorithms such as the LR or the RBFNN, which typically work well with small amounts of data, do not show improved results. However, great benefits can be seen by loosening the tolerances. Thereby, the prediction performance can be increased from ~20% to above 60%, due to the larger amount of data available for training each model. However, the performance remains below the 89% accuracy level, which is achieved for high data volume cutting conditions. This can be explained by the fact that the additional data originates from different cutting conditions. As these strongly influence the investigated cutting forces, it becomes more challenging for the algorithm to learn whether signal deviation is caused by a change in material batch or by a change in cutting conditions. Thus, by increasing the tolerances the deficit of small data can only be compensated partially.

Considering the machinability of the batches, as seen in the Taylor curves for a feed rate of 0.5 mm/rev (Fig. 5a), two clusters of curves can be observed. The lines for batches C, D, and E show similar slopes, while the curve for batch B appears to be decreasing more rapidly. Even though there is some variation among the insert lifetimes for batches C, D, and E, the machinabilities are closer to each other compared to batch B. Therefore, it seems reasonable to consider batches C, D, and E as batches with similar machinability, such that information from the one batch can be applied to the other. Considering this assumption in the prediction matrix,

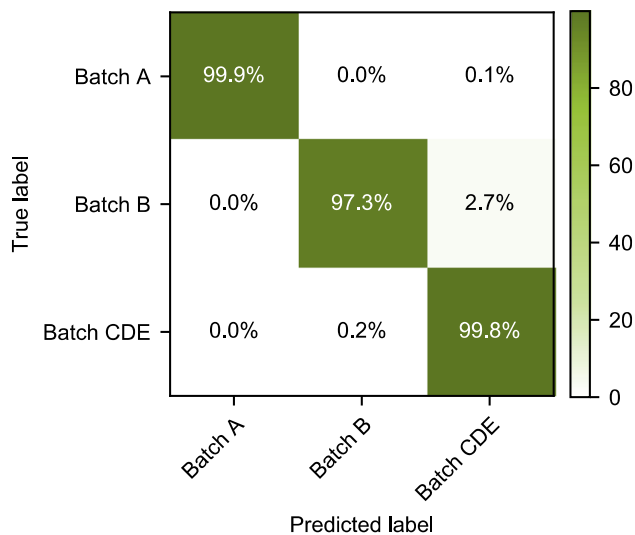


Fig. 8 Assuming batch C, D, and E can be treated as similar due to their close machinabilities, the accuracy of the identification algorithm increases to above 98%

it can be seen that the algorithm nearly perfectly separates the samples of batch A, batch B, and the combined batches C, D, and E, with 99.9%, 97.3%, and 99.8% prediction accuracies, respectively (see Fig. 8). This proves the feasibility of the proposed model, as predictions errors only appear among batches of similar machinability (see Fig. 7).

Lastly, as expected, it can be observed that the computing times increase for higher amounts of data, especially for the best performing algorithm, the SVM. For even larger datasets, a further increase in computation time is expected. This issue can be addressed in operation using two different strategies. One option is to use the best performing but slow-to-train algorithm only if a pretrained model exists, thus no training needs to be carried out during operation. If the material batch must be classified at cutting conditions without a pretrained model, one of the faster models can be trained with the trade-off of lower accuracies. The other option is to outsource the training to cloud computing solutions with higher computing power available, if the training takes too long on local resources.

Conclusion

In this study, a smart service is proposed for identifying the material batch during machining. This information can be used consecutively to optimize the cutting process for each batch individually.

To achieve an in situ material batch identification, the torque values of the machining axes are averaged over a sliding window and evaluated. A database of historic data is filtered by the current cutting conditions for relevant exper-

iments. With the selected data, a ML model can be trained to predict the material batch. For cutting conditions with a low volume of training data, the prediction accuracy can be improved significantly by considering data from nearby, less similar, cutting conditions in the training dataset. However, for cutting conditions with a high volume of training data available, the best results are achieved while only considering data acquired at exactly the same cutting conditions for model training.

Comparing different classification approaches, it is shown that ML outperforms the LR baseline model. The algorithms SVM, RF, and ANN are capable of identifying the material batch with an accuracy of 89.0%, 86.1%, and 85.7% respectively. Analyzing the prediction errors, it is observed that the batches that are misclassified the most are closest to each other in machinability. Treating these batches with similar machinability as a single batch, the prediction accuracy is further increased to 98.9%.

In future research, the proposed smart service has to be adapted for operational needs. Besides implementing the concept as a service-based architecture, strategies for handling an unknown amount of material batches have to be investigated.

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al. (2016). Tensorflow: A system for large-scale machine learning. In *12th symposium on operating systems design and implementation* (pp. 265–283).
- Chollet, F. et al. (2015). Keras. GitHub. <https://github.com/fchollet/keras>.
- Denkena, B., Bergmann, B., & Witt, M. (2018). Automatic process parameter adaption for a hybrid workpiece during cylindrical operations. *The International Journal of Advanced Manufacturing Technology*, 95(1–4), 311–316. <https://doi.org/10.1007/s00170-017-1196-y>.
- Denkena, B., Bergmann, B., & Witt, M. (2019). Material identification based on machine-learning algorithms for hybrid workpieces during cylindrical operations. *Journal of Intelligent Manufacturing*, 30(6), 2449–2456. <https://doi.org/10.1007/s10845-018-1404-0>.

- Dong, Z., Hu, W., & Xue, D. (1994). New production cost-tolerance models for tolerance synthesis. *Journal of Engineering for Industry*, 116(2), 199–206. <https://doi.org/10.1115/1.2901931>.
- Goppold, C., Urlau, F., Pinder, T., Herwig, P., & Lasagni, A. F. (2018). Experimental investigation of cutting performance for different material compositions of Cr/Ni-steel with 1 μm laser radiation. *Journal of Laser Applications*, 30(3), 031501. <https://doi.org/10.2351/1.5013284>.
- Jemielniak, K., & Kosmol, J. (1995). Tool and process monitoring-state of art and future prospects. *Scientific Papers of the Institute of Mechanical Engineering and Automation of the Technical University of Wroclaw*, 61, 90–112.
- Jeon, B., Yoon, J. S., Um, J., & Suh, S. H. (2020). The architecture development of Industry 4.0 compliant smart machine tool system (SMTS). *Journal of Intelligent Manufacturing*. <https://doi.org/10.1007/s10845-020-01539-4>.
- Kibkalt, D., Mayr, A., von Lindenfels, J., & Franke, J. (2018). Towards a data-driven process monitoring for machining operations using the example of electric drive production. In *2018 8th International electric drives production conference (EDPC)* (pp. 1–6). <https://doi.org/10.1109/EDPC.2018.8658293>.
- Knöpfel, A., Gröne, B., & Tabeling, P. (2005). Fundamental modeling concepts. *Effective Communication of IT Systems, England, 2005*, 51.
- Kothuru, A., Nooka, S. P., & Liu, R. (2018). Audio-based tool condition monitoring in milling of the workpiece material with the hardness variation using support vector machines and convolutional neural networks. *Journal of Manufacturing Science and Engineering*, 140(11), 111006. <https://doi.org/10.1115/1.4040874>.
- Kramer, N. (2007). In-process identification of material-properties by acoustic emission signals. *CIRP Annals*, 56(1), 331–334. <https://doi.org/10.1016/j.cirp.2007.05.076>.
- Lutz, B., Kibkalt, D., Regulin, D., Reisch, R., Schiffler, A., & Franke, J. (2019). Evaluation of deep learning for semantic image segmentation in tool condition monitoring. In *2019 18th IEEE international conference on machine learning and applications (ICMLA)* (pp. 2008–2013). <https://doi.org/10.1109/ICMLA.2019.00321>.
- Mayr, A., Kibkalt, D., Meiners, M., Lutz, B., Schäfer, F., Seidel, R., et al. (2019). Machine learning in production—Potentials, challenges and exemplary applications. *Procedia CIRP*, 86, 49–54. <https://doi.org/10.1016/j.procir.2020.01.035>.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: machine learning in python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Penumuru, D. P., Muthuswamy, S., & Karumbu, P. (2020). Identification and classification of materials using machine vision and machine learning in the context of industry 4.0. *Journal of Intelligent Manufacturing*, 31(5), 1229–1241. <https://doi.org/10.1007/s10845-019-01508-6>.
- Schneider, G. (2002). Chapter 3. Machinability of metals. In *Cutting tool applications* (Vol. 67, pp. 2–10). Nelson Pub.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1), 1929–1958.
- Šuchmann, P., & Martinek, P. (2014). Influence of small deviations in steel chemical composition on hard enability. In *Proceedings of materials science and technology 2014* (Vol. 1, pp. 493–499).
- Taylor, F. W. (1906). *On the art of cutting metals* (Vol. 23). American society of Mechanical Engineers.
- Teti, R., Jemielniak, K., O'Donnell, G., & Dornfeld, D. (2010). Advanced monitoring of machining operations. *CIRP Annals*, 59(2), 717–739. <https://doi.org/10.1016/j.cirp.2010.05.010>.
- Teti, R., & La Commare, U. (1992). Cutting conditions and work material state identification through acoustic emission methods. *CIRP Annals*, 41(1), 89–92. [https://doi.org/10.1016/S0007-8506\(07\)61159-7](https://doi.org/10.1016/S0007-8506(07)61159-7).
- Tong, X., Liu, Q., Pi, S., & Xiao, Y. (2020). Real-time machining data application and service based on IMT digital twin. *Journal of Intelligent Manufacturing*, 31(5), 1113–1132. <https://doi.org/10.1007/s10845-019-01500-0>.
- Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data mining: Practical machine learning tools and techniques* (3rd ed.). Burlington, MA: Morgan Kaufmann.
- Yao, X., Zhou, J., Lin, Y., Li, Y., Yu, H., & Liu, Y. (2019). Smart manufacturing based on cyber-physical systems and beyond. *Journal of Intelligent Manufacturing*, 30(8), 2805–2817. <https://doi.org/10.1007/s10845-017-1384-5>.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.