Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 104 (2021) 720–725

www.elsevier.com/locate/procedia

54th CIRP Conference on Manufacturing Systems

Comprehensive machine data acquisition through intelligent parameter identification and assignment

Philipp Gönnheimer [a,*], Andreas Karle[a], Lorenz Mohr[a], Jürgen Fleischer[a]

[a]wbk Institute of Production Science, Karlsruhe Institute of Technology, Kaiserstraße 12, 76131 Karlsruhe, Germany

* Corresponding author. Tel.: +49-1523-950-2578 ; fax: +49-721-608-45005. *E-mail address:* philipp.goennheimer@kit.edu

Abstract

In today's highly competitive manufacturing environment, process data monitoring continues to be of high priority, but often relies on modern communication interfaces being provided by PLC manufacturers. This paper proposes an alternative approach in which data is acquired automatically from various PLC models through available interfaces. Multiple Machine Learning algorithms are incorporated to identify machine parameters, which are then assigned to appropriate machine information models. All functionalities can be provided by a dedicated hardware module or as software modules on IPCs. The proposed approach can be integrated into existing Industry 4.0 efforts to accelerate digitalization in challenging environments.

Keywords: Digital Manufacturing System, Identification, Machine Tool.

1. Introduction

With the goal of increasing the Overall Equipment Effectiveness (OEE) in production, there is an ever-increasing number of approaches and solutions in the area of machine and process monitoring. However, commercially available applications often represent isolated solutions. In addition, there are approaches to process monitoring that are based only on control data such as motor currents and position data, but there is often the difficulty that these machine parameters first have to be extracted and identified from the control system [1]. This is usually a manual and time-consuming process.

In order to automate this process and to make the comprehensive data acquisition for such applications scalable across machines, this paper describes how machine data acquisition can be performed with intelligent parameter identification and assignment.

Nomenclature

OEE	Overall Equipment Effectiveness
PLC	Programmable Logic Controller
SME	Small- and medium-sized Enterprises
ML	Machine Learning
TSC	Time-Series Classification
umati	Universal Machine Technology Interface
FCN	Fully Convolutional Network
ResNet	Residual Neural Network
LSTM	Long Short-Term Memory

2. State of the art and objectives

In this chapter, the state of the art in three fields of research is discussed. First, the retrofitting of existing machinery to integrate I4.0-ready communication capabilities. Second, time-series classification (TSC) through various methods. Third,

10.1016/j.procir.2021.11.121

machine information models. While being subjects of many different research group's active efforts, each of these contribute to the system presented in this paper. The objective, therefore, is to implement these individual components and conduct further research to optimize them for the given application, as described in the fourth part of this chapter.

2.1. Retrofitting

To comply with the I4.0 standard, companies must digitalize their production systems. In particular, small and medium-sized enterprises (SME) are faced with the necessity of integrating modern communication technology with their existing machine inventory. In context of the research project "Retrofitting of machines and plants" (RetroNet), it was determined that the utilization potential of retrofitting might be diminished by its requirements of methodical implementation procedures, additional hardware expenses and qualified personnel on-site [2]. These obstructions are further pronounced if machine controls of several different manufacturers are present. Individual solutions might include third-party modules which facilitate OPC UA communication between machines that might feature proprietary, manufacturer-dependent software [3]. These existing approaches still require an extensive set-up process through qualified personnel, which constitutes a barrier for SMEs when embracing I4.0 developments. The approach presented in this paper attempts to significantly reduce the effort and required expertise to implement these changes.

2.2. Machine information models

A machine information model describes a machine's state with a defined parameter and event structure. Standardization of such models aims to accelerate training of employees and integration into process monitoring systems, facilitating communication between machines of different manufacturers, and general error prevention. The Universal Machine Technology Interface (umati) is a machine information model being developed by the Mechanical Engineering Industry Association (VDMA) and industrial partners on basis of OPC UA since 2018. As of December 2020, the first part, "UA CS for Machine Tools Part 1 - Monitoring and Job", has been made available as OPC UA Companion Specification [4].

2.3. Time series classification

TSC is an established field of research in various disciplines ranging from medicine to climate research [5,6]. A well-established dataset with the purpose of facilitating interdisciplinary cooperation exists in the form of an archive hosted under the cooperation of the University of East Anglia (UEA) and the University of California – Riverside (UCR) [7]. The dataset features a variety of time series data, but none commonly encountered in machine control systems. In recent years, a meta-ensemble of various conventional classification methods called Hierarchical Vote Collective of Transformation-based Ensembles (HIVE-COVE) was developed and constitutes the state-of-the-art solution [8]. Deep Learning methods might offer a more generalized applicability outside the UCR/UEA dataset but have seen only limited research efforts so far [8]. Comparisons of various TSC methods commonly consider accuracy as singular evaluation metric. More recently, researchers achieved valuable results in improving computing efficiency and processing speed [9]. The work presented in this paper aims at developing an accurate, as well as efficient, TSC-method with the specific purpose of identifying machine parameters under on-line processing constraints.

2.4. Research deficits and objectives

None of the existing approaches covers the entirety of the intended application and its challenges – the extraction of data from retrofitted machines, the identification of machine parameters, and the assignment of these parameters to a standardized machine information model under on-line system restraints. The research presented in this paper aims at developing a solution for this key problem and has resulted in several publications so far, which shall be briefly reiterated here. The concept of automated parameter acquisition, identification and assignment to an information model and the results of a probability-based heuristic approach was first published in [10]. A feasibility study of a probability-based heuristic approach identified limitations of that approach. Thereafter, research focus was shifted toward ML-based parameter identification to improve the system's scalability with promising early results [11]. Further work established a multi-stage identification process, incorporating Deep Learning algorithms to successfully identify drive parameters of a limited dataset [12]. This paper discusses the next stage of work. First, an experimental setup is introduced, to apply and evaluate various ML-methods concerning their capacity to identify machine parameters under on-line constraints, with evaluation datasets originating from machines not included in training datasets. Second, the results of in-depth optimization and comparison of several Deep Learning models on basis of an expanded dataset are presented.

3. Machine data acquisition

A comprehensive approach to acquire machine data must accommodate various contingencies. Qualified personnel on-site might not be available. On-line implementation might be required to avoid or minimize machine downtime. Machine control systems might be inaccessible or feature proprietary software restrictions. A varying selection of machine information models might be desired as output. Under these considerations, the target system is best defined as a black box, with as little required user-side input and effort as possible. The black box can be further divided in modules which serve a specific purpose and have defined interfaces, allowing simultaneous developments independent of each other. These modules correspond to the fields of research outlined in the above chapter: data extraction, parameter identification and assignment to a machine information model.

3.1. Experimental setup

The experimental setup consists of two test beds: An I4.0-ready multi-axis milling machine outfitted with Siemens S7 machine controls and Siemens Edge communication modules to accumulate training data and a single-axis horizontal test bed which is mobile for exhibition and demonstration purposes. The test bed features a Bosch Rexroth machine control system which is accessed via OPC UA communication. Several independent research projects concerning leadscrew condition monitoring that require access to machine parameters are implemented on the test bed, serving as a convenient use case for the data acquisition system presented in this paper. ML-algorithms, trained with data acquired from the I4.0 milling machine at high time resolution, are evaluated in their capacity of identifying on-line machine data generated and extracted from the test bed. The training data consists of different milling processes representing the characteristics of seven signal classes on three linear axes and one main spindle. The seven signal classes are binary signals (bin), cycle signals (cyc), control differences (ctrl), currents (cur), torques (tor), the actual and target positions (enc/des pos).

3.2. Process from data extraction to assignment

The procedure to acquire machine data from a machine includes the following steps:

- Determining the presence or absence of a OPC UA server and/or the Programmable Logic Controller (PLC) model
- Establishing a connection to a present OPC UA server or the PLC directly
- Surveying acquisition targets and constraints, such as the number of parameters to identify, the number of machine axes, and achievable query times
- Buffering process data
- Preprocessing and prefiltering of data as required by the specific ML-algorithm
- Identification of machine parameters
- Postprocessing of machine parameters as applicable, such as assignment of dynamic parameters to specific axes
- Assignment of machine parameters to an internal machine information model
- Mapping of input parameters to the machine information model desired by the user, such as umati
- Provision of on-line process data in the desired form on a user-accessible OPC UA server

4. Parameter identification

The studies concerning the parameter identification conducted in this paper can be split in four sections.

First, the impact of two preprocessing parameters on the accuracy of the ML-models have been studied.

Secondly, the ML-models have been embedded in a holistic system, performing a three-stage process to identify all signal classes and assigning them to the different axes

In the third section, the three neural networks (FCN, ResNet and LSTM) as well as a random forest have been optimized and compared to solve the task of parameter identification. The Fully Convolutional Network (FCN) and Residual Neural Network (ResNet) have been chosen due to their recommendation in the work of [13] and the architectures are used without further changes. As recurrent networks are known of their capability of sequential data processing a single-layer Long Short-Term Memory (LSTM) was added for competition and the random forest was expected to be an easy-to-train and transparent model to generate useful insight of the problem.

Conclusively, the final models are compared in a last section and the best models are tested for the purpose of feasible usage on unfamiliar processes and machines.

All four sections are introduced in detail in this chapter and the most important results are discussed in chapter 5.

4.1. Preprocessing parameters

For the time series to be processed, several preprocessing steps must be carried out shown in Fig. 1.

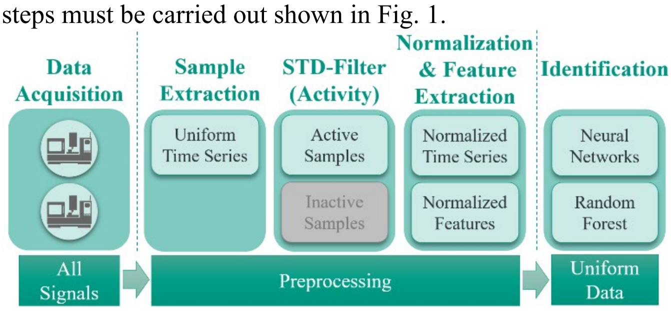

Fig. 1. Steps of the pre-processing in detail

First, samples of specific length must be extracted, which already yields the first preprocessing parameter, the sample length. This preprocessing parameter is measured in data points and regulates the amount of information provided to the models for a single prediction. Desirably this preprocessing parameter should be as small as possible to prevent long data acquisition times and computing costs.

Secondly, the samples must be tested for their activity, meaning that the axes are used in the process at the point of data acquisition. This is requested since inactive signals lead to constant samples causing numeric problems in the following normalization step and contain little to no information about the signal class. Filtering inactive samples can be achieved by calculating the standard deviation (std) of each sample and comparing it with a threshold. This std-threshold is the second preprocessing parameter studied.

As a third step, the samples are standardized, supplying samples with mean of zero and unit variance. This procedure reduces the amount of information in exchange of samples, containing unified signal characteristics which can be translated to different production processes and machine types.

If the chosen model for the parameter identification is a random forest, features must be calculated from the time series to fit the models expected input. Therefore, 40 commonly used features for time series representation are calculated. Those are simple metrics as maximum and minimum and more complex information as deviations and frequencies.

4.2. Parameter identification in three stages

As shown in Fig. 2, the complete process of parameter identification can be split up in three stages.

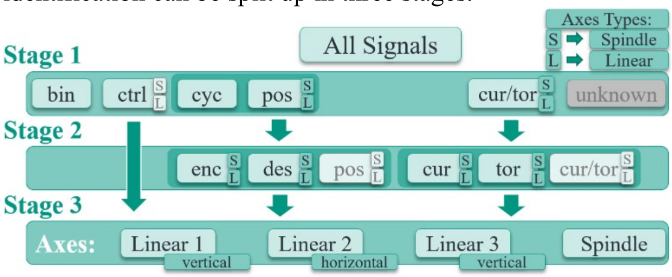

Fig. 2. Three stages of the parameter identification

In stage 1 the selected ML-model predicts a signal class for each signal. Since the neural networks cannot diver between cycle and position signals, a rule-based identification of the cycle based on increase per time step takes place. This rule is reliable due to the specific characteristic of the cycle signal to be incremented by 1 each time step.

To enhance the reliability of the predictions the class probability of each prediction was used to reject uncertain samples. As the class probability strongly depends on the model, the predictions were scanned for an optimal probability threshold to maintain a fair balance of rejecting most wrong samples while remaining as many correct ones as possible.

The following stage 2 aims to expand the position class into actual and target positions signals as well as the current/torque class into separated current and torque signals. Both enhancements can be achieved due to the correlation of the signals within their class and the selection of coupled signals. Since the actual position follows its corresponding target position with a very short time delay, the correlations of the signals are very high. Once all the signal couples are found, a comparison of the local extrema indicates the signals due to the delay.

As the current directly influences the torque and both signals have low resistance to changes, high dependency can be measured for signals of the same axis. The search of a clear rule for the signal identification turned out to be quite challenging and still is unsolved. The signals do not show a time delay and at times reach correlation coefficients of 1, illustrating the similarity of the signals.

Stage 3 maps the signals to their axis due to the dependencies of the signals. As reference for the axes the current/torque class is adduced and then the other signal classes allocated to the axes. The feasibility of this stage was reviewed in [12] and has not been object of further research.

4.3. Hyperparameter optimization of ML-models

The first specification for all models was the number of outputs, representing the signal classes. As shown in [12], training the models to predict fewer signal classes and use expert knowledge as a rule-based subsequence to receive the finale signal classes can be a superior identification process. Afterwards, the hyperparameters of each model have been optimized to provide the fittest algorithms.

The random forest was used to conduct a model-based feature reduction as awareness of highly correlated features existed. Accordingly, the maximum depth of the decision trees and the number of trees in the ensemble was examined. For the neural networks, hyperparameter tuning involved studies of the batch size and training time. Further studies of sample sizes for the LSTM were conducted, due to the unstable convergence with long sample sizes. Finally, the number of hidden states was varied to accomplish a fitting internal complexity.

4.4. Comparison of ML-models and final evaluation

To select the fittest solution, initially the neural networks were compared among each other and the selected model was then matched with the random forest.

As applicability to other production processes and machine tools was of high priority, the winner was trained and tested with separated processes from the test bed to show the capability of the model to transfer learned knowledge to other processes and machine tools.

5. Results and discussion

5.1. Preprocessing parameters

To show the impact of the sample size and std-threshold for the activity filter, random forests were trained to compare the accuracy at specific values of those preprocessing parameters. Especially for the std threshold strong influence on the accuracy was observed. Fig. 3 shows the recalls of the confusion matrix for each output and the accuracy of the model. As the recalls monitor the ratio of true positives to all true samples of each signal class, it is used to measure the model's capability of predicting each class.

The models accuracy rises steadyly with the std-threshold, gaining improvements for two signal classes only. The control difference (ctrl) is calculated based on the position signals (pos) and therefore inherits its dependency from them. Further investigation of position samples with low standard deviation showed high frequent oscilations with low ampplitude, which are not the expected dominant characteristic of such a signal. These oscilations are assumed to be vibrations caused from the milling process. Filtering the inactive samples minimizes the dominance of such vibrations in the models input and consequently reducing misguided classification.

The sample size showed a similar but weaker effect, resulting from longer observations less freqeuently containing no active axis movment. Conclusively, a std-threshold of 0.01

and sample sizes of 1000 data points (2s) are recommended to limit the amount of samples rejected from the filter.

	std-threshold	0	10^-6	10^-5	10^-4	10^-3	10^-2	10^-1	1
recall [%]	binär	99	100	99	100	100	100	100	-
	ctrl	74	70	72	71	49	90	96	-
	cur/tor	97	96	95	93	95	98	99	99
	cyc	100	100	100	100	100	100	100	100
	pos	34	49	49	56	80	90	93	100
accuracy [%]		73	80	80	82	89	96	98	100

Fig. 3. Std-thresholds for filtering inactive samples

5.2. Parameter identification in three stages

Since all stages have been reviewed in [12], this section focuses on the utilization of the model's class probabilities in stage 1 to reject uncertain predictions.

The neural networks already hand out class probability as consequence of their softmax-layer, this information is already available. Decision trees also provide class probabilities as ratios of samples in the leaf of the predicted class and random forests calculate a mean over all trees to generate a final class probability.

To pick a threshold which rejects most false predictions without reducing to many true ones, the number of all predictions are counted for various probabilities, shown in Fig. 4 for the ResNet.

This visualization indicates that a higher percentage of false samples occur within lower class probabilities as correct samples. Due to the general high probabilities of the ResNet a threshold of 0.9999 was selected reducing 47.4% of all wrong classifications while only rejecting 2.8% of the correct predictions.

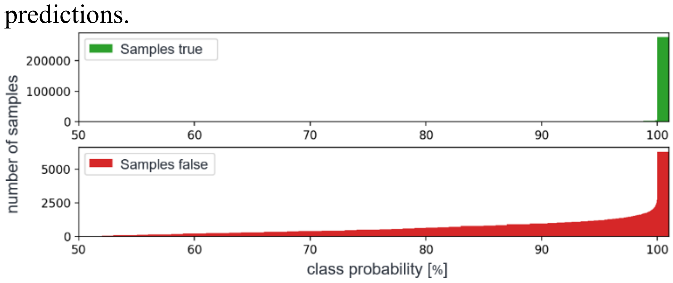

Fig. 4. Number of samples over the class probability for the ResNet

Likewise, this procedure was conducted with the random forest delivering a threshold of 0.6 and resulting in 43.6% less false and 3.6% less correct predictions.

5.3. Hyperparameter optimization of ML-models

To determine the final hyperparameters of the models first the target classes for the classification are presenter and subsequently the specific parameters of the random forest and the neural networks are discussed.

For all models' shortcomings of distinguishing the currents and torques were found and accordingly a common current/torque class was introduced. A similar problem took place when the models where trained to predict the actual and target position, therefore the differentiation of those signal classes is also postponed to stage 2. Further, all neural networks were incapable of separating the positions from the cycles, which the random forest was able to accomplish reliably.

All models could determine the axis type as spindle or linear axis for the current/torque class and the random forest even for its isolated position class. Conclusively, the neural networks classified five signal classes while the random forest could categorize into seven classes.

A recursive feature selection with the permutation importance as a metric to evaluate the feature importance of the random forest showed that with 10 of the 40 features, no decrease of the model's accuracy occurred and therefore only those 10 best features are used to reduce computational costs during the preprocessing. The limitation of a maximum depth showed no advantage of computation time or memory consumption and an amount of 50 learners was found out to be sufficient.

The parameter studies of batch size for the neural networks lead to specific hyperparameters for each net, so the FCN and LSTM were trained with batch size 16 and the ResNet with 64. Since no stable solution with sample size 1000 was found for the LSTM, a length of 100 data points was used. A study of the dimensionality of the hidden state revealed an optimal amount of 200 states and a single layer LSTM was used for the predictions.

5.4. Comparison of ML-models and final evaluation

A comparison of all models trained with their final hyperparameters and 10000 samples, tested on all data, 284400 samples from the I4.0 milling machine, was conducted with the results presented in Fig. 5. The deepest network, the ResNet, demonstrated to have the highest accuracy with 97,8% whereas the FCN showed the highest minimum recall for all classes with 94.6% predicting the spindle axis of the current/torque class.

The random forest achieved a lower accuracy but could classify the position axis types and a separate cycle class.

To finally demonstrate that the models can be used to predict signals from different processes and machines, close to 200 time series of the position, current and torque from the test bed have been extracted. Predicting all data of those processes showed that all signals classes have been predicted with accuracies of 95% by the ResNet and 100% by the random forest. None of the models could distinguish if the axis type of current and torque was linear or spindle. This is assumed to be caused by the different motors of the machine tools and shows that the prediction of the axis type from current and torque signals is less transferable between different machines.

The random forests predict the position signals all correctly as linear axes and a correct classification of the signals due to stage 2 and 3 is possible even without usage of the axis types of the current and torque class.

accuracy/ recall [%]	accuracy	binary	ctrl	cur/tor		pos		cyc
				linear	spindle	linear	spindle	
Random Forest	**94,27**	100,0	93,0	95,5	80,1	99,0	92,3	100,0
ResNet	**97,8**	100,0	97,1	97,5	94,0	99,3		
FCN	**97,4**	100,0	97,1	96,0	94,6	99,4		
LSTM	**89,8**	100,0	93,3	84,6	87,0	93,3		

Fig. 5. Accuracies and class recalls of the ML-models in comparison

6. Conclusion and outlook

The experimental setup was successfully implemented, evaluation datasets were captured and the training dataset already used in previous work was expanded substantially. With this, valuable results could be achieved that will inform further research. The studies of the preprocessing parameters illustrated the importance of the input data quality and that the utilization of a std-filter can improve the model accuracy significantly, due to less noisy data of inactive signals.

Four ML-models have been trained and optimized for the task of parameter identification from time series. The test results indicated that the neural networks could achieve the highest accuracies but, on the downside, did not converge to a minimum in which the positions could be separated from the cycles. In contrast the random forest due to the manually selected features from the preprocessing could reliably predict the cycle classes and since the positions can be isolated the axes types could also be identified.

The successful test with different processes and machine tools demonstrates that the approach and models are mostly transferable. Still, the variety of tested machines is very limited and to identify the applicability to more different machine tools, as turning machines or even robots, more data from other machines is required. The embedding of the ML-models in the multistage system shows how the weaknesses of the ML approach can be compensated with subsequent rule-based expert knowledge. Further, the usage of model-based class probabilities was demonstrated to avoid wrong classifications on the cost of few correct predictions.

Future research efforts will concentrate on further dataset expansion, increasing the number of identifiable parameters, and the practical implementation on various platforms as a black-box system.

Acknowledgements

This publication is based on the research results of the project "EN-AI-BLER - Intelligent Provision of Production Data to Increase Value Creation through AI Applications". The project is funded by the Ministry of Economics, Labor and Housing Baden-Württemberg within the framework of the AI Innovation Competition Baden-Württemberg. The authors of this paper thank the ministry for the funding.

References

[1] Netzer M, Michelberger J, Fleischer J. Intelligent Anomaly Detection of Machine Tools based on Mean Shift Clustering. In: Procedia CIRP. Volume 93; 2020. p. 1448-1453.

[2] Krüger J, Verl A. RetroNet. Retrofitting von Maschinen und Anlagen für die Vernetzung mit Industrie 4.0 Technologie. Dusseldorf: VDI Verlag, 2019

[3] Haskamp H, Orth F, Wermann J, Colombo AW. Implementing an OPC UA interface for legacy PLC-based automation systems using the Azure cloud: An ICPS-architecture with a retrofitted RFID system. In: Proceedings 2018 IEEE Industrial Cyber-Physical Systems (ICPS). Saint Petersburg: IEEE, 2018. p. 115-121.

[4] Goerisch G. OPC UA for Machine Tools – Part 1. Machine Monitoring and Job Overview. 2020. https://opcua.vdma.org/en/catalog-detail/-/catalog/3914

[5] Rajkomar A, Oren E, Chen K, Dai AM, Hajaj N, Hardt M, Liu PJ, Liu X, Marcus J, Sun M, Sundberg P, Yee H, Zhang K, Zhang Y, Flores G, Duggan GE, Irvine J, Le Q, Litsch K, Mossin A, Tansuwan J, Wexler J, Wilson J, Ludwig D, Volchenboum SL, Chou K, Pearson M, Madabushi S, Shah NH, Butte AJ, Howell MD, Cui C, Corrado GS, Dean J. Scalable and accurate deep learning with electronic health records. In: npj Digital Medicine. 2018;1:1-18.

[6] Fulcher BD, Jones NS. Highly Comparative Feature-Based Time-Series Classification. In: IEEE Transactions on Knowledge and Data Engineering no. 26. 2014. p. 3026–3037.

[7] Anh Dau H, Keogh E, Kamgar K, Yeh CCM, Zhu Y, Gharghabi S, Ratanamahatana CA, Chen Y, Hu B, Begum N, Bagnall A, Mueen A, Batista G. The UCR Time Series Classification Archive. 2019. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/

[8] Ismail Fawaz H, Forestier G, Weber J, Idoumghar L, Muller PA. (2019), Deep learning for time series classification: a review. In: Data Mining and Knowledge Discovery. 2019. 33:4-917–963.

[9] Oastler G, Lines J. A Significantly Faster Elastic-Ensemble for Time-Series Classification. In: Yin H. Intelligent data engineering and automated learning, proceedings. Cham: Springer International Publishing. 2019. p. 446-453.

[10] Gönnheimer P, Hillenbrand J, Betz-Mors T, Bischof P, Mohr L, Fleischer J. Auto-configuration of a digital twin for machine tools by intelligent crawling. In: Production at the leading edge of technology. Berlin, Heidelberg: Springer. 2019. p. 543–552.

[11] Gönnheimer P, Netzer M, Mohr L, Hörsten G, Fleischer J. (2020), Erhöhung der Skalierbarkeit von KI-Anwendungen in Produktionsanlagen. In: ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb. 2020.115:7-8-517–519.

[12] Gönnheimer P, Puchta AC, Fleischer J. Automated Identification of Parameters in Control Systems of Machine Tools. In: Production at the leading edge of technology. 2020

[13] Wang, Z.; Yan, W. & Oates, T. (2017 - 2017), „Time series classification from scratch with deep neural networks: A strong baseline“. 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, S. 1578–1585. ISBN: 978-1-5090-6182-2