16th IMEKO TC10 Conference
"Testing, Diagnostics & Inspection as a comprehensive value chain for Quality & Safety
Berlin, Germany, on September 3-4, 2019

Quality Improvement of Milling Processes Using Machine Learning-Algorithms

Maik Frye[1], Robert H. Schmitt[2]

[1]Fraunhofer Institute for Production Technology IPT,
Steinbachstraße 17, Aachen 52074, Germany,
maik.frye@ipt.fraunhofer.de, +49 241 8904-538
[2]Laboratory for Machine Tools WZL RWTH Aachen University,
Cluster Production Engineering 3A 540, Aachen 52074, Germany,
robert.schmitt@rwth-aachen.de, +49 241 80-20283

***Abstract* – The increasing digitalization and industrial efforts towards artificial intelligence foster the use of Machine Learning (ML)-algorithms in the production environment. Within production, different application areas and use-cases arise for the usage of ML. In this paper, we focus on the implementation of ML-algorithms for a milling process where critical process conditions are predicted. Based on the predicted process conditions, the machining parameters can be adjusted in advance to avoid critical conditions of the process. The avoidance of critical process conditions increases the quality of the products, since quality characteristics such as surface roughness or dimensional deviations can be influenced. To ensure the transferability of the results to other applications, we follow a methodical approach. The results of the ML-models are discussed critically and further steps are derived in order to use ML-models successfully in the future.**

Keywords –Quality Improvement, Predictive Process Control, Machine Learning, Artificial Intelligence, Data Preprocessing, Artificial Neural Networks, Random Forest, Gradient Boosting.

I. INTRODUCTION

In recent years, the application of data-driven models (DDM) strongly increased in the production environment. Main reasons for this trend are higher computing power, easier data acquisition, simpler application of DDM and higher reliability of DDM [1]. Currently, 30 % of German companies use ML for process optimization [2]. According to the study, the application of ML in production will double by 2023. This trend will be further intensified due to the "Artificial Intelligence (AI)" research year proclaimed by the German government, which opens up a wide range of applications areas for ML in production [3]. Application areas range from Predictive Maintenance (PdM) to optimization of routing and scheduling to Predictive Process Control (PPC). Even process as well as product design can be influenced by applying ML-algorithms. In this paper, we focus on PPC and, more specifically, on a milling process of high precision and highly stressed components used in aerospace industry. A major challenge during milling pose the emergence of critical process conditions such as vibrations, high forces and accelerations, which have a negative impact on the component surface quality and, thus, on the product quality. Since the machining parameters such as axis data and spindle speed influence critical process conditions, data is acquired, which is analysed in order to gain insights of the process and, in a second step, to predict critical process conditions by applying ML-algorithms.

The accurate prediction of critical process conditions can be used to adjust machining parameters in advance so that critical process situations do not even occur. Based on the ML-results, machining parameters can be adjusted. The machining parameter adaptation requires a further model, which in practice are heuristics. In this work, the focus is not on heuristics, but rather lies on the evaluation of suitable ML-algorithms that are able to predict critical process parameters, in this case the magnitude of the amplitude of vibrations, accurately.

This work differs from previous IMEKO-publications, which have already focused on ML. So far, results have mainly been published which concentrate on the entire production or individual components [4–6]. In addition, further IMEKO-publications from the past years are related to other fields of application such as electrical engineering [7, 8]. In the present case, reference is made to a single milling process and thus to mechanical engineering. Moreover, this publication can be assigned to the research topic "Data Analytics, artificial intelligence techniques and machine learning for testing, diagnostics and inspection" (T5).

II. RELATED RESULTS IN THE LITERATURE

Related results in the literature refer mainly to the prediction of product and process quality [9]. An overview of existing use-cases shows the great potential of ML algorithms in production. [10, 11]

Krauß et al. (2019) focused on the prediction of product quality based on a process chain consisting of multiple processes. For each process, a classification and regression tree (CART)-algorithm was trained in order to classify whether the corresponding product would be in or out of specification at the end of the process chain. The CART-algorithm was evaluated based on the Matthews correlation coefficient and achieved a performance of MCC = 0.70. [12]

Kim (2018, p. 556) showed that most ML-studies are focusing on the prediction of tool wear, surface roughness as well as process parameter in milling processes. In this context, the most common ML-algorithms are artificial neural networks (ANN), support vector machines (SVM) followed by decision trees. [11]

Krishnakumar et al. (2015) conducted experiments in order to classify the cutting tool condition in milling of titanium alloys. For this purpose, the ML-algorithms decision tree and artificial neural network (ANN) are applied. The results reveal almost identical performance of the decision tree and ANN. Both algorithms predicted the tool condition of 87 samples with a classification accuracy of about ACC = 90 % [13]. Recently, Saadallah et al. (2018) applied ML-algorithms for time series data to predict the stability of a milling process. In addition, simulation data was used to train an ensemble of deep learning (DL)-algorithms to realise a live recommendation system. [9]

Regarding milling and surface finish predictions, Venkata Rao et al. (2014) used a feed forward neural network to predict tool wear, amplitude of work piece vibration and workpiece vibration [14]. Wu et al. (2018) applied random forests to predict tool wear in a milling process for a condition monitoring system. Statistics of cutting forces, vibrations and acoustic emissions where fed to a random forest (RF) of 10,000 trees. [15]

It can be concluded that ML is already used in the production area, even if its use is limited to prototypes and pilots. In the development of pilots, however, the focus is on the optimization of production processes such as milling. However, the data basis is not sufficient to achieve good generalisation results in the vast majority. Moreover, the performance of ML-algorithms are yet not evaluated on the milling of high precision and highly stressed components used in the aerospace industry.

III. DESCRIPTION OF THE METHODOLOGY

As in many cases within production, the methodology utilised in this work is inspired by cross-industry standard process for data mining (CRISP-DM)-framework [16]. Figure 1 shows the consecutive stages of the derived methodology. Exemplarily, the methodology is displayed only with two ML-algorithms. In the first step, data of the milling process is acquired. The next stage comprises the data preprocessing, in which the data is cleaned, synchronized, augmented and features are selected.

Subsequently, ML-algorithms are selected, which includes the derivation of requirements from the data set characteristics and requirements of the milling process. After implementation, the selected ML-algorithms are validated. Finally, all results are evaluated and compared. In the following, each step of the methodology is presented in detail.

A. Process Understanding

During milling, two separate data sets are acquired with different sample frequencies but the same time stamp. The first data set comprises the machine data including the coordinates of the 5-axis milling machine as well as the spindle speed, feed and torque, while the second data set has two attributes consisting of the sound pressure level with corresponding time stamps.

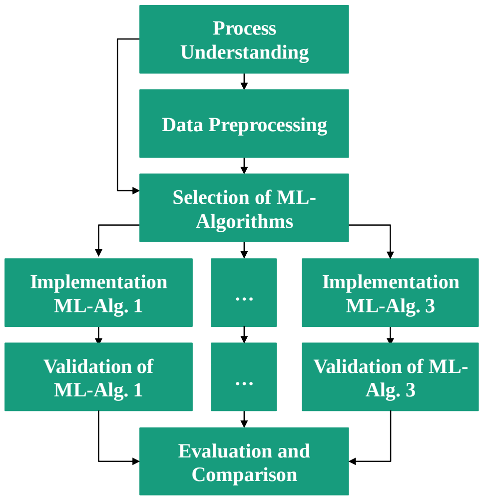

Figure 1. Methodology of ML-implementation

Table 1 summarizes the characteristics of the data set. The number of observations in the machine data set is less than 1 % of the amount of observation of the microphone data. A microphone is mounted within the machine in order to record the sound pressure level (SPL). By recording the pressure level over time, a Fast Fourier Transformation (FFT) can be applied to obtain the amplitude and frequency of the vibration.

B. Data Preprocessing

In a first step, both data sets are integrated into one data base. Because of different sample frequencies, the data sets need to be synchronized. In practice, data synchronization pose a major challenge when synchronizing different data sources. In addition to the different frequency, a different staring time of data acquisition can also occur. The machine data is assigned to the vibration data with the corresponding time stamp. Since it is assumed that the machine data change linearly, missing values are augmented by using the imputation technique linear regression.

Before selecting appropriate ML-algorithms, features of the data sets are engineered. Firstly, the feed of the spindle is removed since the values remain constant at 100 % all the time. Next, the vibration data is modified through mapping of all data points to the absolute value.

In addition, the signal of the SPL was further engineered. The SPL of the current and the past two instances were used as an input of the ML-algorithm. In addition, the first derivative and second derivative of the SPL of the current and past instance were calculated to determine the change and slope of the signal. Outliers in the data are automatically removed and all features are transformed through standardisation. Since the ML-output serves as input for the heuristic, which triggers the adjustment of process parameter during machining, a prediction horizon H needs to be calculated. The prediction horizon includes the calculation time of the ML-algorithm, the calculation time of the heuristic, the signal processing of the machine as well as the actual physical change of the parameter in the machine. Based on a comprehensive literature research, the prediction horizon H is set as H = 184.255 ms.

Table 1. Data set characteristics.

	Machine data set	Microphone data set
No. of observations	12,875	21,720,837
No. of attributes	8	2
Milling Time [s]	424	424

C. ML-Algorithm Selection

Since all the data is labelled and the goal is to predict the magnitude of amplitudes, we applied supervised learning-algorithms with the regression learning task. The ML-algorithms are selected on the basis of the requirements of the milling process, the properties of the data set, external requirements such as computing time and the successful ML-application in similar use-cases. These approach leads to the selection of RF, gradient boosting and Long Short-Term Memory (LSTM) neural networks.

RF is a tree-based algorithm with many single predictors, whose separate predictions are aggregated to one superior prediction [17]. A RF utilises a group of decision trees that are built separately while including random operations in training. Therefore, Bootstrap Aggregating (bagging) is performed to build the ensemble learners [18]. By applying bagging, training data sets are selected randomly for the respective decision tree. Each bootstrap comprises the same amount of instances such as the original data set [19]. The decision trees are trained using the Classification and Regression Tree (CART)-algorithm. When training is finished, prediction of decision trees are aggregated. [20]

Gradient boosting is also a tree-based algorithm. First, one decision tree is trained based on the training data set. Afterwards, a second decision tree is trained to predict the residuals of the first one. Subsequently, a third decision tree is trained to predict the residuals of the second decision tree. This iterative process can be extended arbitrarily. In the end, all prediction are summed up. [21]

LSTM neural networks are a special form of the Recurrent Neural Network (RNN). RNN are capable of modelling specially sequential data through the inclusion of connections that are pointing backwards to neurons of previous layers or to the input of the same neuron. Thus, timely dependencies are incorporated in the architecture [17]. An issue with RNNs are exploding and, more likely, vanishing gradients [22]. In case of exploding gradients, weights between hidden layers may become very big during backpropagation, which let the ML-algorithm diverge. Through limiting the norm of gradients, the exploding gradients can be mitigated [22]. This technique is called gradient clipping. When gradients approximate to the norm zero, gradients are vanishing, which lead to the fact that the early layers in the network will not contribute to learning, since the gradient tend to become smaller and vanish while propagating backwards during training. [23] As a solution of vanishing gradient problem in RNN, Hochreiter & Schmidhuber suggested the LSTM cell. Instead of a neuron with an activation function such as rectified linear unit (ReLU), a LSTM cell consists of several activation functions and signal flows. [24]

D. ML-Algorithm Implementation

ML-algorithms can be applied using either open-source or proprietary platforms. Generally, proprietary platforms like Matlab are useful for fast model development, however, licences can lead to high operational costs. On the other hand, there are open source platforms such as Python or R, which are not only free of charge but also plays a dominant role in recent ML-developments. Thus, Python including the existing libraries such as scikit-learn for RF as well as gradient boosting and TensorFlow for LSTM neural networks are used. The computing operations were conducted applying a computing cluster that consists of 64 cores and an internal memory of 256 Gigabyte.

Two RF are trained with 20 decision trees and 40 decision trees respectively. The decision trees are fully grown and not pruned. Moreover, two gradient boosting algorithms are trained. The number of predictors are 50 and 100. Studies by Hastie et al. (2017) lead to the setting of the learning rate of $\eta = 0.1$ and the number of leafs $n_{leaf} = 8$ [25]. Based on the findings of Friedman (2002), a random sub sample size of 40 % is applied for training the predictors. [26]

Two LSTM neural networks are implemented, consisting of two hidden layers, 20 LSTM-cells per layer, a batch size of batch = 32 and the learning rate of $\eta = 0.001$ [27]. In both cases, the gradient optimizer Adam is used. The neural networks are trained over 50 epochs. According to Srivastava et al. (2014), one neural network is regularized based with a dropout rate of $p = 0.5$ as well as the max-norm regularization with a norm boundary for weight vectors of $C = 4$. [28]

E. Validation and Evaluation

The validation of the ML-algorithms are conducted based on forward chaining cross validation. The data set is split into six different folds as indicated in the literature and common in comparable use-cases [29]. The folds of the data set, which are located after the test fold, are ignored in order to allow realistic training scenarios. Since the learning task is a regression, the performance of the ML-algorithms are evaluated based on the root mean square error (RMSE). In the end, results are compared.

IV. RESULTS AND DISCUSSIONS

Table 2 shows all ML-algorithms results ranked by the validation error. The regularized LSTM-neural network has the lowest validation error of RMSE = 0.2920. There are differences in training and validation error. Contrary to what has been proven in the literature, validation errors are not always higher than training errors [30]. For instance, gradient boosting and LSTM neural networks have lower validation than training errors. RFs achieve the lowest training error while having the highest validation error.

Table 2. Results of Ml-algorithms.

	RMSE (validation)	RMSE (training)
LSTM neural net (incl. regularization)	0.2920	0.3890
Gradient boosting (100 predictors)	0.3089	0.4628
Gradient boosting (50 predictors)	0.3144	0.4765
LSTM neural net (excl. regularization)	0.3248	0.3800
Random forest (40 predictors)	0.3269	0.1478
Random forest (20 predictors)	0.3329	0.1549

Although RFs performed worst, peaks in the time series were approximated the best (see Figure 2). Figure 2 shows the comparison between actual amplitudes and the amplitudes predicted by the 40-tree RF exemplarily for all ML-algorithms. In case of RFs, the predictions on the validation set reflect a high validation error and poor generalisation of the trained algorithm. The RF underestimated the real values, which originates from averaging the single predictors. The differences to the real values are much higher for predictions on the validation set. According to Géron (2017, p. 127), a low training error together with a high validation error indicates overfitting [17]. Since the majority of data is actually background noise, the algorithm learns primarily to predict the low amplitudes (93.64 % of the data) during training. RFs of fully-grown trees overfit on noisy data sets.

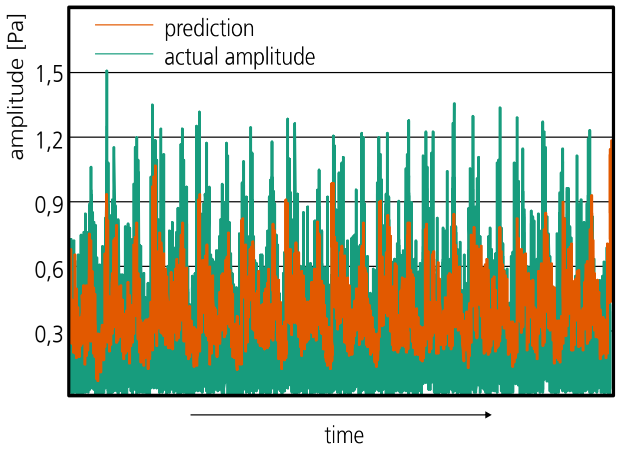

Figure 2: Validation results of RF with 40 predictors

For every ML-algorithm, the learning curves are calculated in order to identify when the ML-algorithm converge. Figure 3 shows the learning curve of the RF with 40 predictors. It can be seen that the training error continues decreasing but tends to start converging up from around 15 decision trees. The same principle can be taken from the validation error, which stays on a level of around RMSE = 0.33 Pa after approximately ten decision trees. In Table 2, it was shown that the RMSE-values for both training and validation of RF with 20 and 40 predictors do not differ significantly through the early convergence. Adding more predictors does not contribute new information to the model.

In addition, the relative importance of the feature are recorded. The features of the machine axes y and a as well as the second derivative of the amplitude of the sound pressure level deliver most information for the RF. Additional time series features contribute around 53 % to the prediction. The training time was $t_{train} = 28.50$ h by utilising 30 processor cores.

The results of the gradient boosting algorithms are characterized by higher training than validation errors (see Table 2), which is not expected from literature [30]. The feature importance and the predictions reveal that both algorithms only consider axis data. The reason is the limited selection of features, since the training of the predictors is stopped after three splits. Furthermore, the overall vibration is decreasing over time. Based on the chosen validation approach and the fact that the algorithms predicts noise of the data, the algorithm achieve better results during validation than training. The same principle can be observed at LSTM neural networks. Here, the algorithm with the lowest validation error was already reached after the first epoch of training, which means that the models start to overfit directly.

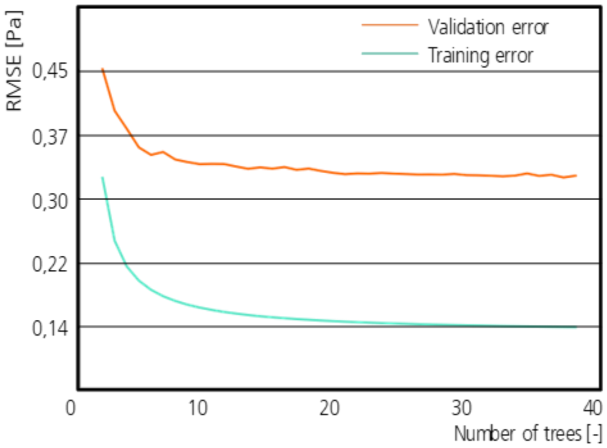

Figure 3: Learning curve of RF with 40 predictors

Based on the results, fields of action can be derived. In the first step, the data quality needs to be enhanced through the elimination of noise. In following works, different noise filter methods will be tested in order to reduce the noise in a proper manner. Three different ensemble and vote-based methods will be implemented such as Ensemble Filter [31], the Cross-Validated Committees Filter [32] and the Iterative-Partitioning Filter [33].

In order to deploy an automated system that controls process parameters, new data has to be prepared automatically and standardised to feed the ML-models. When feeding the ML-model with new data automatically, the data preprocessing need to be automated and a data pipeline needs to be created in order to stream data. Thus, further research need to ensure that the process of streaming and preprocessing data for ML-application for the given use-case is stable.

The focus of further research regarding ML-models lays in methods to adjust hyperparameters. Recently published hyperparameter methods such as BOHB seem to outperform basic approaches such as random search in terms of accuracy and advanced approaches like Bayesian optimization in terms of computational efforts [34]. Applying these methods will lead to higher accuracy of ML-models.

In addition, research has to determine how to change process parameters depending on the outcome of the forecast. Based on the predictions of the ML-model, heuristics can be used to adjust process parameters in order to stay within tolerance after the end of the process.

V. CONCLUSIONS AND OUTLOOK

Critical components in the aerospace industry have to meet high requirements regarding product quality characteristics. In order to improve product quality, critical process conditions during milling processes must be avoided. Parameter optimisation during the milling process serves as an opportunity for avoiding critical process conditions. Following a methodological approach, the initially acquired data from different data sources needed to be preprocessed. For this reason, data synchronization was performed and most relevant features were engineered. Based on the derived requirements, three suitable ML-algorithms were investigated to forecast vibrations during milling. The ML-algorithms were the decision tree-based algorithms RF and gradient boosting as well as LSTM neural network. By applying forward chaining cross validation and by calculating the RMSE-errors, the ML-algorithms were validated and evaluated.

The results show that in the course of this work, a RF approximates high amplitudes better than gradient boosting and LSTM neural networks. The reasons can be traced back to the selected hyperparameters of the chosen algorithms. Aside from the detection of peaks, the regularized LSTM achieved the lowest validation error.

Further improvements can be achieved by investigating ways of handling noisy data, searching for optimal hyperparameter settings, establishing standardised and automated data preprocessing methods and developing models for process parameter adjustments based on predictions of the ML-algorithm. In the context of production processes, in which especially critical components of aeronautical engineering are manufactured and in which ML-algorithms are applied at the same time, certification aspects have to be considered. For this reason, interest groups are already dealing with this topic in order to determine guidelines for certification. This can lead to a successful deployment of ML-algorithms to improve the quality in milling processes.

VI. ACKNOWLEDGMENTS

The European Union under H2020-Widespread 2016/2017 "Teaming for Excellence" sponsors this project to promote further development of data analytics and machine learning with the granted number 739592. Research Executive Agency (REA) supervises the development of the project under the power delegated by the European Commission.

VII Literatur

[1] **Driscoll, M.:** Building data startups: Fast, big, and focused. Low costs and cloud tools are empowering new data startups, 2011

[2] **Geissbauer, R.; Schrauf, S.; Berttram, P.; Cheraghi, F.:**

Digital Factories 2020. Shaping the future of manufacturing, https://www.pwc.de/de/digitale-transformation/digital-factories-2020-shaping-the-future-of-manufacturing.pdf., 2017

[3] **Bundesministerium für Bildung und Forschung**: Wissenschaft im Dialog: Wissenschaftsjahr 2019 - Künstliche Intelligenz, https://www.wissenschaftsjahr.de/2019/.

[4] **Viharos, Z. J.; Csanaki, J.; Nacsa, J.; Edelényi, M.; Péntek, C.; Balázs Kis, K.; Fodor, A.; Csempesz, J.:** Production trend identification and forecast for shop-floor business intelligence. ACTA IMEKO 5, No. 4, 2016, pp. 300-306.

[5] **Capriglione, D.; Carratù, M.; Sommella, P. et al.:** ANN-based IFD in Motorcycle Rear Suspension., https://www.imeko.org/publications/tc10-2017/IMEKO-TC10-2017-002.pdf., 2017.

[6] **Presenti Campagnoni, V.; Ierace, S.; Floreani, F.:** A diagnostic tool for Condition-Based Maintenance of circuit breaker, https://www.imeko.org/publications/tc10-2016/IMEKO-TC10-2016-026.pdf, 2016.

[7] **Polok, B.; Bilski, P.:** Optimization of the neural RBF classifier for the diagnostics of electronic circuit, https://www.imeko.org/publications/tc10-2017/IMEKO-TC10-2017-020.pdf., 2017.

[8] **Bilski, P.:** Unsupervised learning-based hierarchical diagnostics of analog circuit, https://www.imeko.org/publications/tc10-2017/IMEKO-TC10-2017-016.pdf., 2017.

[9] **Saadallah, A., Finkeldey, F., Morik, K., Wiederkehr, P.:** Stability prediction in milling processes using a simulation-based machine learning approach. In: Procedia CIRP, 2018, pp. 1493–1498.

[10] **Köksal, G., Batmaz, I. and Testik, M. C.:** A review of data mining applications for quality improvement in manufacturing industry. In: Expert Systems with Applications, 2011, pp. 13448–13467.

[11] **Kim, D.-H., Kim, T. J. Y., Wang, X., Kim, M., Quan, Y.-J., Oh, J. W., Min, S.-H.:** Smart Machining Process Using Machine Learning: A Review and Perspective on Machining Industry. In: International Journal of Precision Engineering and Manufacturing-Green Technology, 2018, pp. 555–568.

[12] **Krauß; J., Döhler Beck, G. T., Frye, M., Schmitt, R. H.:** Selection and Application of Machine Learning-Algorithms in Production Quality. In: Niggemann, O.; Kühnert, C.; Beyerer, J. (Hrsg.): Machine Learning for Cyber Physical Systems, Springer-Verlag, 2019, pp. 46–57.

[13] **Krishnakumar, P., Rameshkumar, K., Ramachandran, K. I.**: Tool Wear Condition Prediction Using Vibration Signals in High Speed Machining (HSM) of Titanium (Ti-6Al-4V) Alloy. In: Procedia Computer Science, 2015, pp. 270–275.

[14] **Venkata K., Murthy, B., Mohan N.:** Prediction of cutting tool wear, surface roughness and vibration of work piece in boring of aisi 316 steel with artificial neural network. In: Measurement, 2014, pp. 63-70.

[15] **Wu, D., Jennings, C., Terpenny, J., Kumara, S. and Gao, R. X.:** Cloud-based parallel machine learning for tool wear prediction. In: Manufacturing Science and Engineering 2018.

[16] **Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., Wirth, R.:** CRISP-DM. Step-by-step data mining guide, 1999.

[17] **Géron, A.:** Hands-on machine learning with Scikit-Learn and TensorFlow. Concepts, tools, and techniques to build intelligent systems. Sebastopol: O'Reilly Media, 2017.

[18] **Breiman, L.:** Random Forests, In: Machine Learning, Volume 45, 2001, pp. 5–32.

[19] **Pham, D. T.; Afify, A. A.:** Machine-learning techniques and their applications in manufacturing. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture 219, No. 5, 2005, pp. 395–412.

[20] **Han, J., Kamber, M., Pei, J.:** Data mining: Concepts and techniques. The Morgan Kaufmann series in data management systems. Amsterdam: Elsevier/Morgan Kaufmann, 2012.

[21] **Friedman, J. H.:** Greedy function approximation: A gradient boosting machine. In: The Annals of Statistics, pp. 1189–1232, 2001

[22] **Goodfellow, I., Bengio, Y. and Courville, A.:** Deep learning. London: MIT Press 2016.

[23] **Pascanu, R.; Mikolov, T.; Bengio, Y.:** On the difficulty of training Recurrent Neural Networks, 2012.

[24] **Hochreiter, S.; Schmidhuber, J.:** Long Short-Term Memory. Neural Computation 9, No. 8, 1997, pp. 1735–1780.

[25] **Hastie, T., Tibshirani, R., Friedman, J. H.:** The elements of statistical learning: Data mining, inference, and prediction. In: Springer series in statistics. New York: Springer 2017.

[26] **Friedman, J. H.:** Stochastic gradient boosting. In: Computational Statistics & Data Analysis - Nonlinear methods and data mining, 2002, pp. 37–378.

[27] **Greff, K., Srivastava, K., Koutnik, J., Steunebrink, R., Schmidhuber, J.:** LSTM: A Search Space Odyssey. In: IEEE transactions on neural networks and learning systems, 2017, pp. 2222–2232.

[28] **Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.:** Dropout a simple way to prevent neural networks from overfitting. In: Journal of Machine Learning Research, 2014, pp. 1929–1958.

[29] **Bergmeir, C.; Costantini, M.; Benítez, J. M.:** On the usefulness of cross-validation for directional forecast evaluation. Computational Statistics & Data Analysis 76, 2014, pp. 132–143.

[30] **Witten, I. H. and Frank, E.:** Data mining: Practical machine learning tools and techniques. Amsterdam and Boston: Morgan Kaufman, 2005.

[31] **Batista, G. E. A. P. A.; Prati, R. C.; Monard, M. C.:** A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsletter 6 (2004), No. 1, p. 20.

[32] **Kibler, D.; Aha, D. W.:** Learning Representative Exemplars of Concepts: An Initial Case Study. In: Proceedings of the Fourth International Workshop on MACHINE LEARNING, Elsevier, 1987, pp. 24–30.

[33] **Devijver, P. A.:** On the editing rate of the Multiedit algorithm. Pattern Recognition Letters 4, 1986, pp. 9–12.

[34] **Falkner, S.; Klein, A.; Hutter, F.:** BOHB: Robust and Efficient Hyperparameter Optimization at Scale 2018, http://arxiv.org/pdf/1807.01774v1, 2018.