



A Holistic Approach to Part Quality Prediction in Injection Molding Based on Machine Learning

Alexander Schulze Struchtrup^(✉), Dimitri Kvaktun,
and Reinhard Schiffers

Universität Duisburg-Essen, Lotharstraße, 1, 47057 Duisburg, Germany
{alexander.schulze-struchtrup,dimitri.kvaktun,
reinhard.schiffers}@uni-due.de

Abstract. All plastics processing companies have to fulfill the objectives of time, cost and quality. Against this background, those producing in high wage countries are especially challenged, because superior part quality is often the only possibility to prevail in competition. Since this leads to high expenses on quality assurance, for some time already efforts have been made to predict the quality of injection molded parts from process data using machine learning algorithms. However, these did not yet prevail in industry, mainly for two reasons: First, because of the inevitable learning effort that is required to set up a quality prediction model and second, because of the complexity in the application. Current research in the field of transfer learning aiming to shorten learning phases addresses the first challenge. In this paper, we present a holistic approach for the data analysis steps that are necessary once process and quality data have been generated, aiming to minimize the application effort for the operator. This includes the development and application of suitable algorithms for automatic selection of data, process features as well as machine learning algorithms including hyper-parameter optimization and model adaption. Combining the two approaches could bring quality prediction one significant step forward to successful industry application. Beyond this, the presented approach is universally applicable and can therefore be used for other plastics processing methods as well.

Keywords: Injection molding · Quality prediction · Machine learning

1 Introduction

Thermoplastics injection molding is a discontinuous process that allows the automatic, highly reproducible production of molded parts with complex geometry [1–3]. Over the last couple of decades, injection molding machines have been improved with regard to both, mechanical precision and control techniques [4]. Anyway, internal and

external perturbations, such as viscosity fluctuations of the melt, may negatively affect the quality of the molded parts.

Consequently, plastics processing companies are spending large efforts on quality assurance. Nevertheless, scrap production is usually detected only with delay in a sample based quality inspection, while bad parts produced between two samples may stay completely unnoticed. To overcome these drawbacks, research efforts have been made, to predict the quality of the molded parts directly from machine and process data using machine learning algorithms [5–11].

Despite good results, these approaches did not prevail in industry to date, although corresponding products [12] are available. In the authors' perception, this is mainly due to two drawbacks: First, because of the obligatory learning phase during which a quality prediction is not yet possible. Second, since the process of robustly building a good quality model requires many steps. These include data generation and selection, feature extraction, construction and selection as well as learning and adapting suitable models including hyperparameter-optimization, which, in the past approaches had to be carried out mainly manually and resulted in a lot of effort.

The first issue is addressed in recent research [13–17] dealing with the transfer of relationships learned from simulation data as well as other molded parts, aiming to shorten the learning phase of the new model. Still, there is no holistic approach that holistically analyzes, combines and automatically carries out the previously named data analysis steps (cf. Fig. 1) in the context of injection molding quality prediction. Therefore, we present such an approach for injection molding quality prediction and share some results of our research in the first three named areas.

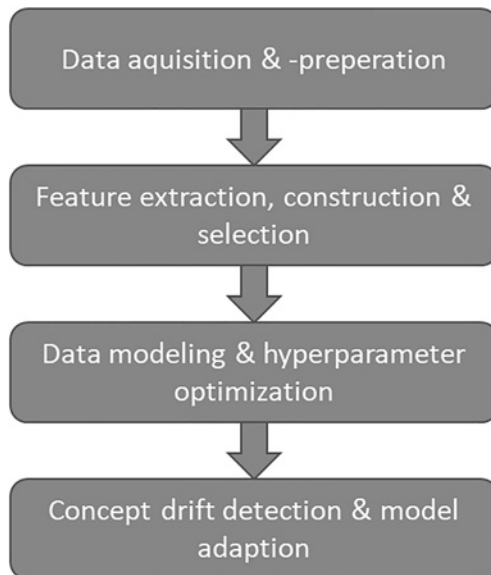


Fig. 1. General framework for holistic quality prediction.

2 Methods

2.1 Data Acquisition and Preparation

The experiments were carried out on a KraussMaffei 120-380 PX fully electric injection molding machine (IMM) in a production cell with a linear robot, conveyor belt and 100% inline quality monitoring. The machine has standard sensor technology with two additional cavity pressure sensors in the mold, which are directly connected to the machine data processing system. Six different experiments were conducted: stable process, start-up process, downtime process¹, process with re-grind material,² process with re-grind material and adaptive process control (APC) from KraussMaffei and a design of experiment (DOE). Therein, injection velocity, the holding pressure, the holding pressure time, cooling time and the barrel temperature at the nozzle were varied (cf. Table 1), creating 43 different combinations.

Table 1. Process setup for DOE. For all other experiments, the machine setting parameters were set to the central point of the DOE.

	Lower axial point	Lower cube point	Center point	Upper cube point	Upper axial point
Injection velocity [mm/s]	72	80	100	120	125
Holding pressure [bar]	429	450	500	550	571
Holding time [s]	2,9	3,5	5,0	6,5	7,1
Cooling time [s]	9	10	12	14	15
Nozzle temperature [°C]	226	230	240	250	254

Each experiment consists of 1000 injection molding cycles creating 1000 data samples, except the DOE with 860 cycles and 860 data samples, respectively. The weight and length of the molded parts (rectangular plate specimen, cf. Fig. 2) were measured directly after every completed injection cycle. The process and quality data were interfaced and evaluated with Matlab 2019a. In total, 48 machine and process parameters and two corresponding quality criteria were logged during each cycle.

After the data acquisition, the data needs to be prepared for the future steps like feature selection and data modeling. In general, the data is split into two parts. The first part is used for training the model and adjusting the (hyper-) parameters. The second part, the validation set, is used to estimate the generalization error of the model. The objective is a good prediction with a low generalization error. This method is called cross-validation [21]. In this work 80% of the data is used for training and 20% for validation, which is a subgroup of cross-validation called holdout method [22].

¹ In total 9 downtimes of 5, 15 and 25 min (3 times each), taking place every 100 cycles.

² Variation of re-grind material fraction from 0 to 100% in steps of 25%, 200 cycles each. Used material: Polypropylene LyondellBasell Moplen HP501H.

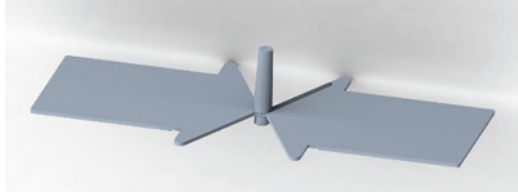


Fig. 2. Plate specimen used for the data generation experiments.

2.2 Feature Selection

The quality of the prediction model depends on the amount and quality of the data as well as the input features used for the modeling [18]. Most often, parameters are chosen through trial and error or expert knowledge [19, 20]. Since this work pursues a holistic approach for quality prediction, the parameters are chosen automatically using state of the art feature selection methods, while the resulting model quality is compared. Feature selection algorithms can be divided into three types. The first method is called embedded method where the feature selection is part of the learning process. Wrappers are the second method where the predictor which is used as a black box is tested with different subsets of features, trying to improve the overall prediction performance. Filter methods are the last approach of feature selection which are independent of the predictor. The selection is done directly by some performance evaluation metrics (PEM). Filter methods are usually less computationally expensive than embedded or wrapper methods [18], which is why they are mainly used in this work.

Search Strategies. Even when using computationally efficient feature selection methods such as filters, it may still be not feasible to evaluate every possible feature subset. Therefore, search strategies are applied, which yield still good results while minimizing the required computational resources.

In a forward selection (FS) the algorithm starts with an empty feature set and continuously adds features trying to improve the PEM. In a backward elimination (BS) the procedure starts with all features and progressively deletes the feature, which is least useful regarding the PEM [18]. Although they are computationally very efficient, both suffer from the “nesting effect”. It describes the case that features which are selected through the FS, cannot be discarded later while features which are discarded in the BS cannot be re-selected [23].

A solution for this problem are floating search methods. The sequential floating forward selection (SFFS) starts with an empty feature set. In the first step the normal FS algorithm is applied and one feature is added to the feature set. In the second step one feature is conditionally excluded applying the normal BS. If this new subset is the best so far, the conditionally excluded feature is removed from the feature set and the algorithm starts with step 2 again. If the subset is not the best so far, the conditionally excluded feature is returned to the feature set and the algorithm continues with step 1 [24]. The sequential floating backward selection (SFBS) is the opposite of the SFFS and starts with all features in the feature set. In the first step the normal BS algorithm is applied and the least significant feature is excluded from the feature set.

In the second step one discarded feature is temporarily added to the feature set applying the normal FS algorithm. If the new subset gives the best PEM, the temporarily included feature is added to the feature set and the algorithm continues with step 2. If the subset is not the best so far, the feature is not added and the algorithm continues with step 1 [23].

Performance Evaluation Metric. With the performance evaluation metric, the significance of a feature is evaluated. In this work the Correlation-based Feature Selection (CFS) according to HALL [25] is selected:

$$M_s = \frac{k\bar{r}_{cf}}{\sqrt{k + k(k-1)\bar{r}_{ff}}} \quad (1)$$

where k is the number of features in the subset, \bar{r}_{cf} is the average of the correlations (relevance criterion) between the features and the class (quality criterion), \bar{r}_{ff} is the average feature-feature inter-correlation and M_s is the resulting PEM merit [25]. According to HALL “a good feature set is one that contains features highly correlated with the class, yet uncorrelated with each other”. Other PEMs like Relief [26], minimum redundancy – maximum relevance [27] or mutual information [28] are beyond the scope of this paper. Figure 3 shows the feature selection process.

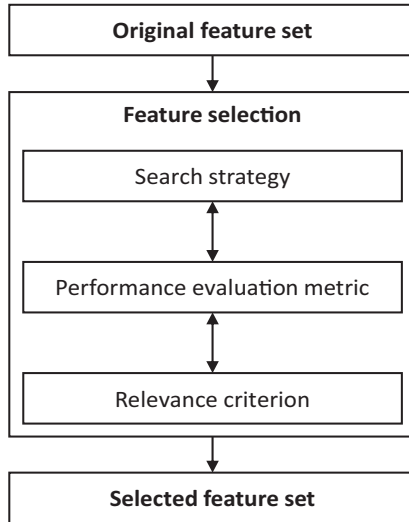


Fig. 3. Feature selection with interactions of search strategy, PEM and relevance criterion.

2.3 Data Modeling and Hyperparameter-Optimization

Machine learning methods can be distinguished in three main classes: supervised, unsupervised and reinforcement learning. All machine learning methods used in this work are supervised machine learning methods. In supervised learning the predictor

learns the relation between the inputs and outputs [29]. Furthermore, supervised machine learning can be separated in two classes depending on the output data type. If the output data is discrete the problem is called classification, if the output data is continuous the problem is called regression [30]. Since the weight and the length of the component are continuous, the machine learning algorithms used in this work are those suitable for regression problems. The following six machine learning algorithms are used: Artificial neural networks (ANN) [31], support-vector machines [32], binary decision trees [33], k-nearest-neighbors (kNN) [34], ensemble methods (LSBoost [35] & random forest [36]) and Gaussian process regression [37]. Furthermore, normal multiple linear regression [38] is added to the analysis to compare classical statistical methods with machine learning.

Every machine learning method has so called hyperparameters that need to be set by the user to maximize the effectiveness of the machine learning method. They are used to define numerous configurations of the algorithm affecting both learning process and the resulting model structure. Examples are the number of neurons in the hidden layer of an ANN or the number of neighbors in the kNN-method. Most frequently, hyperparameters are set via rules-of thumb, by testing sets on a predefined grid or by the default configuration of the software-provider. In this paper the hyperparameter-optimization is done by Bayesian optimization, which proved to be a very efficient method with good performance [39]. Table 2 provides an overview over the hyperparameters chosen for optimization.

Table 2. All machine learning methods tested in this work with their hyperparameters. In total 22 predictors were learnt for every experiment and both quality key figures.

Algorithm	Hyperparameter
Multiple linear regression	–
Artificial neural networks	<ul style="list-style-type: none"> • Number of neurons in the hidden layer • Transfer function • Learning rate • Momentum [40]
Support-vector machine	<ul style="list-style-type: none"> • Penalty factor • Epsilon • Kernel function • Polynomial order [41]
Binary decision trees	<ul style="list-style-type: none"> • Maximum number of splits • Minimum leaf size [35, 42]
k-nearest neighbors	<ul style="list-style-type: none"> • Number of neighbors • Distance metric [43]
Ensemble-method	<ul style="list-style-type: none"> • Method • Number of learning cycles [35]
Gaussian process regression	<ul style="list-style-type: none"> • Kernel function • Kernel scale • Sigma [37]

3 Evaluation

3.1 Data Generation

One objective of this study is to evaluate the six different experiments, which represent possible process states occurring in real-world injection molding production. Figure 4 shows the best possible result of the 22 different predictors for the six experiments in regard of the two quality parameters. The coefficient of determination is used to evaluate the models' prediction quality on the validation dataset.

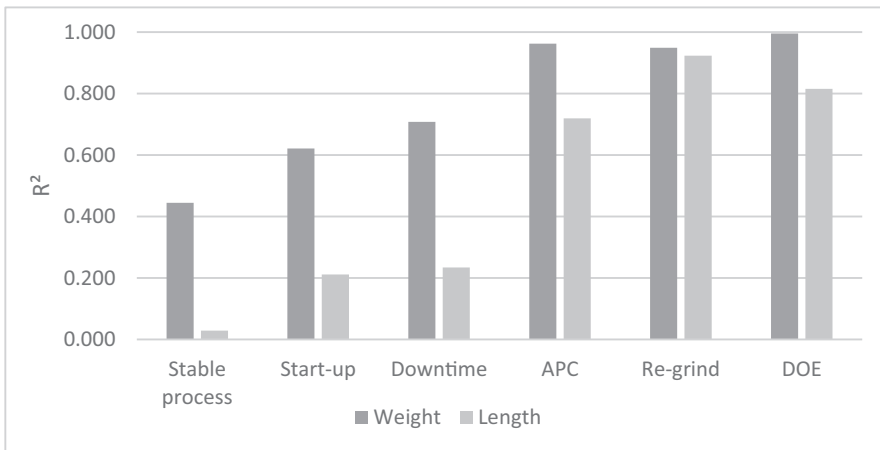


Fig. 4. Comparison of the prediction quality for the different experiments.

It can be seen that the DOE, the process with re-grind material and APC and the process with only re-grind material are easier to predict than the stable process, start-up and downtime. The best prediction result provides the DOE for the weight with a R^2 of 0.995, i.e. 99.5% of the weight variance can be explained by the model. In general, it can be observed that the weight is easier to predict than the length. An explanation could be that the measuring accuracy of the weight measurement compared to the systematical variation of the quality criterion is higher than that of the length measurement.³ A greater measuring effort is likely to improve the model quality for the length prediction as well. It is also apparent that process situations with a low variance in the process parameters e.g. stable process are harder to predict than process situation with high variance e.g. DOE (cf. Table 3). This might be one reason, why APC, re-grind and DOE yield better results than the stable process having the smallest standard deviations.

³ The parts were weighed with a Sartorius Entris 153I-1S balance with 1 mg reproducibility, part length were extracted algorithmically from images taken with a Canon Eos 5D Mark III DSLR with EF 70-200mm f/4L USM objective.

Table 3. Standard deviations (Mean values for weight and length are approximately 19.7 g and 182.7 mm respectively) of weight and length from the different experiments

Experiment	Weight [g]	Length [mm]
Stable process	0.005	0.023
Start-up	0.010	0.027
Downtime	0.015	0.025
APC	0.028	0.048
Re-grind	0.029	0.094
DOE	0.277	0.060

In the coming sections, only the results for the weight will be shown, as the length prediction shows qualitatively comparable behavior and a complete presentation (weight and length) would exceed the scope of this paper.

3.2 Feature Selection

Figure 5 shows the coefficient of determination of the individual feature selection methods for each experiment. More precisely, the best learning method with the respected R^2 was selected. E.g. the best R^2 for the wrapper approach is 0.334 for the stable process which was achieved by the ensemble predictor.

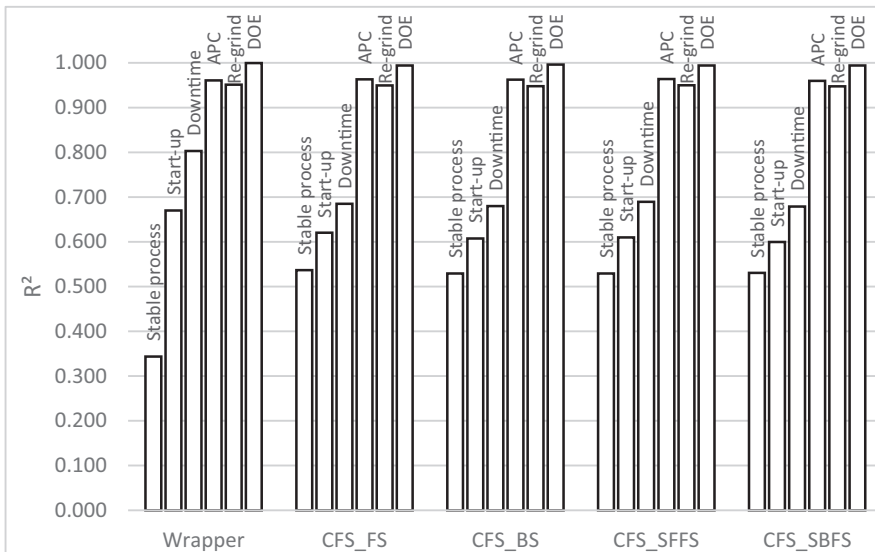


Fig. 5. Comparison of the predictive performance for the different feature selection algorithms in terms of weight.

From the comparison of feature selection methods, one can generally derive two conclusions: First, the results of the wrapper approach using multiple linear regression differs from the results of the CFS filters and second, the different search strategies only slightly affect the filters' performances. While the wrapper yields better results on the start-up and downtime datasets, the CFS performs better on the stable process data. On the other three datasets, no significant differences occur.

The wrapper was only tested with linear regression as predictor, using the FS as search strategy. The selected features with this method were used for the other predictors acting as a filter method [18]. Figure 5 also shows that the experiments with a high variance (cf. Table 3) are easier to predict than the experiments with low variance. While the features, which are selected, vary, but the overall performance is within a similar range.

Figure 6 exemplarily shows the prediction performance of the wrapper depending on the number of features selected for the re-grind experiment. The R^2 for the training dataset is most of the time higher than for the validation dataset, which was to be expected since the training data is known to the predictor while the validation set is not. The highest R^2 is reached for 13 features with a value of 0.9165. For higher feature numbers, the training R^2 continues to increase, while the test R^2 is decreasing due to overfitting.

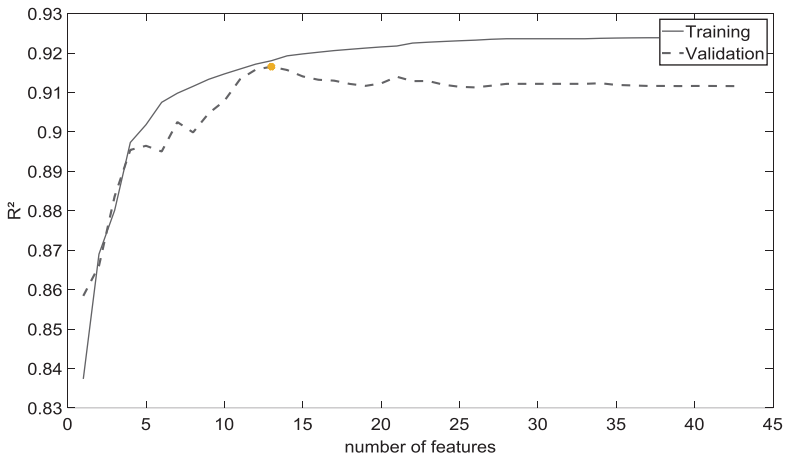


Fig. 6. Prediction quality for the weight using the wrapper approach. Coefficient of determination in terms of the number of selected features for the training and validation re-grind dataset.

3.3 Learning Algorithms

After evaluating the influence of process states and feature selection algorithms on model quality, we now want to compare the learning algorithms themselves.

As it is possible to see from Fig. 7 the Gaussian process regression is the best predictor for the prediction of the weight in every experiment. The overall highest

coefficient of determination is 0.995 for the DOE with the Gaussian process regression. Like the results shown in Fig. 5 the DOE, the process with re-grind material and the process with re-grind material and APC are good to predict. Furthermore, the Gaussian process regression stands out on the stable-process data, a dataset with small variance all other algorithms have trouble to predict. It might also surprise, that multiple linear regression does not perform worst at all on all datasets: despite its simple model structure, especially on the start-up and downtime data it exceeds expectations and yield above average results. Besides kNN, which yields mainly below average results, the other algorithms (ANNs, SVM, decision trees and the ensemble) have a generally comparable predictive quality.

In general, it becomes obvious, that the process state used for data generation is much more important than the learning machine, since the algorithm can only extract correlations that are present in the data.

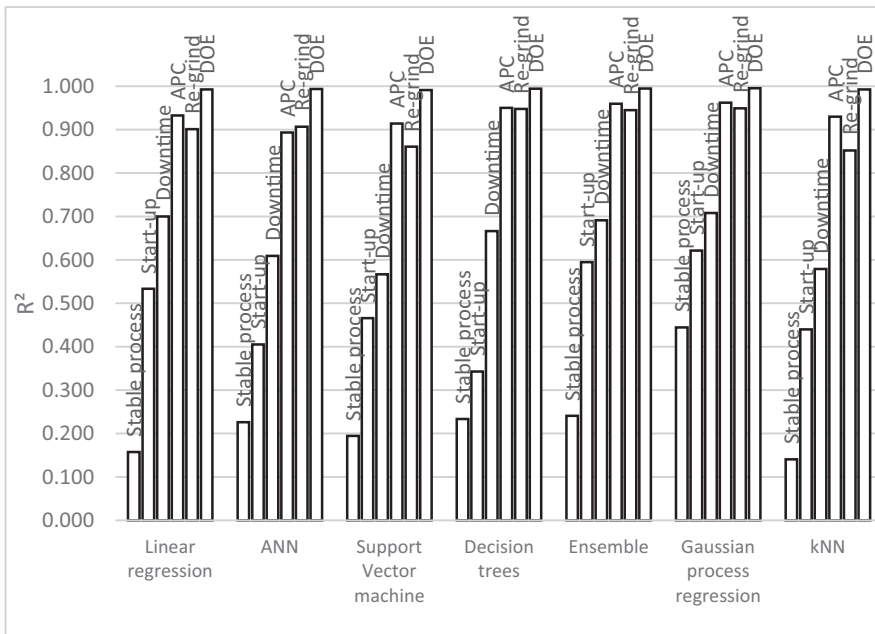


Fig. 7. Coefficient of determination for the different learning algorithms. The weight should be predicted.

4 Conclusion and Outlook

In this study six different experiments were done using a KraussMaffei 120-380 PX injection molding machine. The data include 48 machine and process parameters as well as the weight and the length of the molded parts as quality criteria. 1000 molding cycles (860 for the DOE) were carried out. The pre-processing of the data included

cross-validation using 80% of the data for training and 20% for validation of the models. In the first step feature selection was executed, comparing a wrapper approach with four filter methods. The filter methods contained FS, BS, SFFS and SBFS as search strategies. The PEM was CFS according to HALL with the Pearson correlation coefficient. In total 22 predictor models were built and their hyperparameters were optimized using the Bayesian optimization. Six machine learning methods, including ANN, Support-vector machine, Decision trees, Ensemble, Gaussian process regression, kNN and normal linear regression were compared. The prediction performance of the different models was calculated through the coefficient of determination.

The results show, that process states with a high variance of the quality criteria, such as those based on the variation of the re-grind material fraction and the DOE provide the best base for learning good quality prediction models. The weight is better predictable than the length, with the highest R^2 of 0.995 for the DOE learned by a Gaussian process regression, which yielded the best results on the other datasets as well. Regarding the evaluated feature selection methods, their influence on the model quality was rather small when comparing the different search strategies. However, the presented wrapper's and filters' performances differed significantly on three out of six datasets. Still, it is hard to judge which approach is better, since there is no method outperforming the others on all datasets. Additionally, other feature selection methods might perform different, so in future work different PEMs should be addressed like mutual information or Relief. Furthermore, the framework should be expanded to other machine learning methods in particular methods for classification. Also, a holistic approach has to deal with the detection and reaction to concept drift, which might negatively affect the predictive quality.

References

1. Hopmann, C., Michaeli, W.: Einführung in die Kunststoffverarbeitung, 7th edn. Hanser, Munich (2015)
2. Hopmann, C., Michaeli, W., Greiff, H., et al.: Technologie des Spritzgießens, 4th edn. Hanser, Munich (2017)
3. Standard DIN 24450: Maschinen zum Verarbeiten von Kunststoffen und Kautschuk. Beuth, Berlin (1987)
4. Schiffers, R.: Verbesserung der Prozessfähigkeit beim Spritzgießen durch Nutzung von Prozessdaten und eine neuartige Schneckenhubführung. PhD thesis (2009)
5. Gierth, M.: Methoden und Hilfsmittel zur prozessnahen Qualitätssicherung beim Spritzgießen von Thermoplasten. PhD thesis (1992)
6. Hanning, D.: Continuous Process Control. Qualitätssicherung im Kunststoffverarbeitungsprozess auf Basis statistischer Prozessmodelle. PhD thesis (1994)
7. Häußler, J.: Eine Qualitätssicherungsstrategie für die Kunststoffverarbeitung auf der Basis künstlicher Neuronaler Netzwerke. PhD thesis (1994)
8. Vaculik, R.: Regelung der Formteilqualität beim Spritzgießen auf Basis statistischer Prozessmodelle. PhD thesis (1996)
9. Al-Haj Mustafa, M.: Modellbasierte Ansätze zur Qualitätsregelung beim Kunststoff-spritzgießen. PhD thesis (2000)

10. Schnerr, O.: Automatisierung der Online-Qualitätsüberwachung beim Kunststoffspritzgießen. PhD thesis (2000)
11. Walter, A.: Methoden des prozessnahen Qualitätsmanagements in der Kunststoffverarbeitung. PhD thesis (2000)
12. Liedl, P., Haag, G., Müller, H., et al.: Spitzenqualität mit kurzen Zyklen. *Kunststoffe* **2**, 38–40 (2010)
13. Hopmann, C., Theunissen, M., Heinisch, J.: Von der Simulation in die Maschine – objektivierte Prozesseinrichtung durch maschinelles Lernen. In: VDI Jahrestagung Spritzgießen, Baden-Baden (2018)
14. Hopmann, C., Theunissen, M., Wipperfürth, J., et al.: Prozesseinrichtung durch maschinelles Lernen. *Kunststoffe* **6**, 36–41 (2018)
15. Hopmann, C., Wahle, J., Theunissen, M., et al.: Flexibilisierung der Spritzgießfertigung durch Digitalisierung. In: *Kunststoffindustrie 4.0 – 29. Internationales Kolloquium Kunststofftechnik*, pp. 76–88 (2018)
16. Tercan, H., Guajardo, A., Heinisch, J., et al.: Transfer-learning: bridging the gap between real and simulation data for machine learning in injection molding. *Procedia CIRP* **72**, 185–190 (2018)
17. Hopmann, C., Bibow, P., Heinisch, J.: *Internet of Plastics Processing*. IPC Madison, USA (2019)
18. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *J. Mach. Learn. Res.* **3**(3), 1157–1182 (2003)
19. Charest, M., Finn, R.; Dubay, R.: Integration of artificial intelligence in an injection molding process for on-line process parameter adjustment. In: *Annual IEEE International Systems Conference (SysCon)*, pp. 1–6. IEEE, Vancouver, Canada (2018)
20. Gao, H., Zhang, Y., Zhou, X., Li, D.: Intelligent methods for the process parameter determination of plastic injection molding. *Front. Mech. Eng.* **13**(1), 85–95 (2018)
21. Duda, R., Hart, P., Stork, D.: *Pattern Classification*, 2nd edn. Wiley, New York (2001)
22. Arlot, S., Celisse, A.: A survey of cross-validation procedures for model selection. *Stat. Surv.* **4**, 40–79 (2010)
23. Pudil, P., Novovičová, J., Kittler, J.: Floating search methods in feature selection. *Pattern Recogn. Lett.* **15**(11), 1119–1125 (1994)
24. Chandrashekar, G., Sahin, F.: A survey on feature selection methods. *Comput. Electr. Eng.* **40**(1), 16–28 (2014)
25. Hall, M.A.: *Correlation-based feature selection for machine learning*. PhD thesis (1999)
26. Kira, K., Rendell, L.A.: The feature selection problem: traditional methods and a new Algorithm. In: *AAAI'92 Proceedings of the Tenth National Conference on artificial Intelligence*, pp. 129–134. AAAI, San Jose, California, USA (1992)
27. Ding, C., Peng, H.: Minimum redundancy feature selection from microarray gene expression data. In: *IEEE Computer Society Bioinformatics Conference*, pp. 523–528, IEEE, Stanford, USA (2003)
28. Hall, M. A., Smith, L. A.: Practical feature subset selection for machine learning. In: *ACSC'98 Proceedings of the 21st Australasian Computer Science Conference*, pp. 181–191. ACSC, Perth, Australia (1998)
29. Russell, S.J., Norvig, P.: *Artificial intelligence*, 2nd edn. Prentice Hall, Pearson Education, Upper Saddle River (2003)
30. Alpaydin, E.: *Introduction to machine learning*, 2nd edn. MIT Press, Cambridge (2010)
31. Hagan, M.T., Demuth, H.B., Beale, M.H.: *Neural Network Design*, 1st edn. PWS, Boston (1996)
32. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. *Stat. Comput.* **14**(3), 199–222 (2004)

33. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and Regression Trees, 1st edn. CRC Press, Boca Raton (1984)
34. Biau, G., Devroye, L., Dujmović, V., Krzyżak, A.: An affine invariant k-nearest neighbor regression estimate. *J. Multivar. Anal.* **112**, 24–34 (2012)
35. Hastie, T., Tibshirani, R., Friedman, J.H.: The Elements of Statistical Learning. Data Mining, Inference, and Prediction, 2nd edn. Springer, New York (2017)
36. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001)
37. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning, 3rd edn. MIT Press, Cambridge (2008)
38. Urban, D., Mayerl, J.: Angewandte Regressionsanalyse: Theorie, Technik und Praxis, 5th edn. Springer VS, Wiesbaden (2018)
39. Claesens, M., De Moor, B.: Hyperparameter search in machine learning. In: MIC 2015: The XI Metaheuristics International Conference, pp. 1–5, MIC, Agadir, Morocco (2015)
40. Bengio, Y.: Practical recommendations for gradient-based training of deep architectures. *Lecture Notes in Computer Science 7700 LECTURE NO*, pp. 437–478 (2012)
41. Ito, K., Nakano, R.: Optimizing Support Vector regression hyperparameters based on cross-validation. In: Proceedings of the International Joint Conference on Neural Networks, pp. 2077–2082. IEEE, Portland, USA (2003)
42. Matignon, R.: Data Mining using SAS Enterprise Miner, 1st edn. Wiley-Interscience, Hoboken (2007)
43. Wilson, D.R., Martinez, T.R.: Improved heterogeneous distance functions. *J. Artif. Intell. Res.* **6**, 1–34 (1997)