Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 00 (2021) 000–000

www.elsevier.com/locate/procedia

54th CIRP Conference on Manufacturing Systems

Predictive analytics in quality assurance for assembly processes: lessons learned from a case study at an industry 4.0 demonstration cell

Peter Burggräf[a], Johannes Wagner[a], Benjamin Koke[a], Fabian Steinberg[a], Alejandro R. Pérez M.*[a], Lennart Schmallenbach[a], Jochen Garcke[b,c], Daniela Steffes-lai[b], Moritz Wolter[b,d]

[a] *Chair of International Production Engineering and Management (IPEM), Universität Siegen, Paul-Bonatz-Straße 9-11, Siegen - 57076, Germany*
[b] *Fraunhofer Institute for Algorithms and Scientific Computing (SCAI), Schloss Birlinghoven 1, Sankt Augustin- 53757, Germany*
[c] *Institut for Numerical Simulation, Universität Bonn, Endenicher Allee 19b, 53115 Bonn*
[d] *Institut for Computer Science, Universität Bonn, Endenicher Allee 19a, 53115 Bonn*

* Corresponding author. Tel.: +49-271-740-4509; fax: +49-271-740-2630. *E-mail address:* alejandro.perez@uni-siegen.de

Abstract

Quality assurance (QA) is an important task in manufacturing to assess whether products meet their specifications. However, QA might be expensive, time-consuming, incomplete, or delayed. This paper presents a solution for predictive analytics in QA based on machine sensor values during production while employing machine-learning models based on logistic regression in a controlled environment. Furthermore, we present lessons learned while implementing this model, which helps to reduce complexity in further industrial applications. The paper's outcome proves that the developed model was able to predict product quality, as well as to identify the correlation between machine-status and faulty product occurrence.

© 2021 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)
Peer-review under responsibility of the scientific committee of the 54th CIRP Conference on Manufacturing System.

Keywords: Machine-Learning; Predictive Quality; Production; Quality Assurance; Logistic Regression

1. Introduction

For companies aiming at quality leadership, the final quality of any goods produced is vital for market success. Thus, it is highly important for companies to ensure that every product meets the expected customer and legal requirements. The quality control (QC) of production processes and products is one fundamental part of general quality management. It contains the measures and activities needed to fulfill specific quality requirements [14].

Over the years, diverse methods to ensure the quality of products have been developed [14]. Among the most used methods, we have Auditing, Total Quality Control (TCQ) [8], Poka-Yoke Techniques [16], the 100% revision of each produced product [7], and statistical methods like the Statistical Process Control (SPC) [19]. Independently of the method used, the goal is the same, identify and removed any defect or faulty pieces on the produced products before sending them to costumers [14]. However, the relationship between cost-efficiency vary from one method to another. This is noticeable, e.g., at the cost of highly specialized equipment for the products evaluation, or the cost of employes performing a manual inspection of the produced products versus the accuracy of detecting all faulty products [17].

In general, the quality characteristics of a product can be classified as variables or attributes. "Attributes" are qualitative measurable characteristics, i.e, color, texture, other. On the other hand, "variables" are quantitative measurable characteristics, which are shown as numbers [14]. The advantage of variable measurements is that in the event of a defective intermediate or end product, precise measured values are available, which enable precise adjustment of the process parameters, as well as the possibility of observing trends in the collected data. This would enable the possibility to employ predictive models to determine the quality of the product before the QC process takes place [3, 10].

With the evolution of machine learning (ML) applications [6], related works which combines QC and predictive models are becoming more relevant. The applications diversify from

2212-8271 © 2021 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)
Peer-review under responsibility of the scientific committee of the 54th CIRP Conference on Manufacturing System.

the analysis of the produced piece in search for defects [18, 20], to process oriented QC [11, 10, 2, 15].

Considering the high amount of data available at today industries, an approach to improve the existent relationship between cost-efficiency and QC accuracy would be to implement a ML model capable to predict the product quality based on the initial state of the machine. This will not only help to detect faulty products before the QC process is done, it would also strengthen the current QC methods by combining them with QC predictive models as to compensate their current weakness. While implementing ML models to QC, several challenges arise that have not yet been solved. In this paper, we address the following questions:

- Where to test predictive analytics models in a controlled industrial environment?
- What to consider when evaluating the output from ML models for industrial applications? Accuracy vs Applicability
- What steps should be follow as to prepared the data for predictive QC methods?
- What findings and lessons are worth to consider for further applications in the area?

For this, we cover the methods used in quality check in production, as well as the bases of predictive analytics in production (Section. 2). We describe a suitable insdustrial platform to implement the ML models (Section. 3). Then, we explains the three machine learning methods implemented on this work: Multilayer perceptrons, Support Vector Machines, Decision Tree (Section. 4), followed by the description of the experiment process from the data acquisition and preprocessing, to the classifier optimization and testing (Section. 5). Finally we describe the remarks and discoveries achieved after the finalization of this project phase, as well as presents a list of lessons learned during the implementation of the project, which might be helpful on further implementations (Section. 6).

2. Related Works

2.1. Quality control in production

In this paper, we focused on the product-oriented methods at the product level.

The Statistical Process Control (SPC) method is normally used to monitor the production process using recorded data values for the quality characteristics, as well as to indicate any significant changes in the quality characteristics in the production process. SPC methods can also take place before the actual manufacturing process, this is commonly known as ”off-line SPC” [19]. During the design phase of a process or product, off-line SPC procedures are deployed. These procedures aim to increase the quality of the outcoming product by choosing conforming products and process parameters [14]. The biggest advantage of working with SPC is that large batches can be checked without major time impact on the QC process, however, this would be only possible if the variance of the process product parameters is small enough, to ensure process capability [9]. It is important to remark that statistical methods such as acceptance sampling or quality control charts represent a compromise between zero and the hundred percent control [19, 12]. Which means, if a batch is labeled as NOK, there is no specific reference of when did the error occurs and how many pieces are affected.

Another conservative technique to perform the quality check would be to check 100% of the produced parts. In this method, every product is controlled and sorted at designated quality levels. However, this has the disadvantage of high cost related to personal or specialized material. Generally, this method is used when the consequences of letting a defective item through could be severe, e.g., for specialized equipment or when the cost of delivering faulty products is very high [7].

The quality characteristics of a product can be classified as variables or attributes. Attributes are qualitative measurable characteristics, i.e, color, texture, other. Variables are quantitative measurable characteristics, which are shown as numbers [14]. The advantage of variable measurements is that in the event of a defective intermediate or end product, precise measured values are available, which enable precise adjustment of the process parameters, as well as the possibility of observing trends in the collected data. This would enable the possibility to employ predictive models to determine the quality of the product before the QC process takes place.

2.2. Predictive analytics in production

According to Delen and Damirkan, predictive analytics (PA) consists of unraveling the inherent relationships (if any) between input and output by using data and mathematical techniques [4]. PA offers a wide range of applications, and it can be implemented, as long as the provided data are sufficiently available [3]. According to Krauß, there is a wide range of applications for PA in the industrial and production environment. Some common industrial applications of PA in the industry are focused products (design and optimization), machines and assets (predictive maintenance, anomaly detection, self learning-machines) and process (scheduling, process design, predictive process control)[10]

Applications that combine PA with QC might improve the current QC processes. By employing machine learning (ML) techniques, see Sec. 4, together with traditional methods, it would be possible to upgrade the currents QC methods, and improve the cost-efficiency relationship in the QC process, as well as, decrease the risk of labeling NOK pieces as OK and deliver them to costumers.

Based on an explorative literature review on this matter, the short outcome is that the findings that combine PA and QC are very scarce in the literature. This could be in great part, due to computer power limitations in early research and the recent increase of ML applications in this area. Most of the related works found, tends to be from recent years [11, 10, 2, 18, 20, 15].

3. The case industry 4.0 demonstration cell

Our case of study is an abstraction of an assembly process of the industry. The industry 4.0 demonstration cell is composed of three independent conveyor belts, a robotic assembly arm (UR3), a laser scanner used for quality control as well as a wide range of sensors, all orchestrated by a SIEMENS PLC S7-1200, see Figure 1. The executed abstracted assembly process consists of stacking two disks of different sizes on top of each other. These pseudo product quality are later determined by the laser scanner by evaluating the concentricity of both disks. If the disks' concentricity is within a tolerance of 1.5mm, the piece is classified as OK, if not, it is classified as NOK. By considering that malfunctions can occur during industrial production processes, our demonstration cell can simulate failures in diverse areas, such as:

- Assembly errors due to the robotic arm: incorrectly positioned disks.
- Bearing damage on the conveyor belts: vibration changes in the conveyor belt motors.
- Resistance on the conveyor belts: temperature changes in the conveyor belt motors.
- Malfunction of the gate door due to leakage in the compressed air system: cylinder stroke at the assembly barrier is slower.
- Faster operating point: disks shift position due to the high belt speeds.
- Missing material at the end of the warehouse: production stop due to lack of material.

Those simulated failures have an impact in the data collected by the sensors. Anomalies can be observed at the vibration sensors, temperature rise, fluctuations in the air pressure system, noticeable shifting on pieces' position as well as the stop of the process.

The continuous data collection and the clear correlation between the machine parameters is the ideal play-ground to implement a sandbox for machine learning implementations. With this in mind, our goal is to predict the quality of the product (concentricity of the assembled parts), prior to the actual quality check done by the laser scanner.

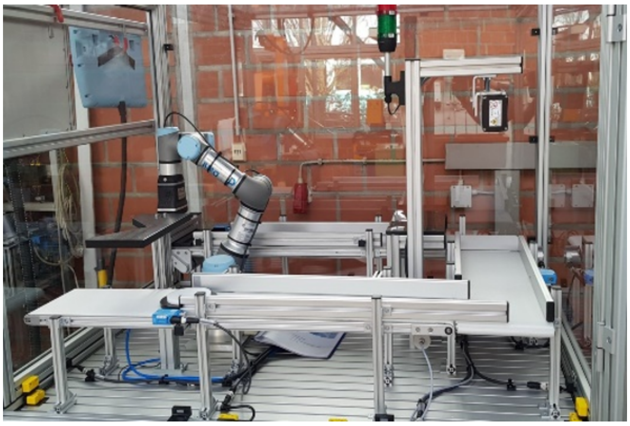

Fig. 1: The industry 4.0 demonstration cell at the University of Siegen

4. Machine Learning Methods

To study which implementation would be more suitable for industrial applications, we compare Multilinear Perceptrons (MLPs), Support Vector Machines (SVM), as well as random forests on the task of quality prediction. All three will be introduced next.

4.1. Multilayer perceptrons

These feedforward neural networks typically combine multiple layers and activation functions [3]. A layer contains a large weight matrix and a bias vector. To evaluate the layer the input vector must be multiplied with the weight matrix before the bias vector is added. Each layer is typically followed by an activation function which adds non-linearity to the network graph. Rectified linear units (ReLUs) are zero for negative inputs leave positive inputs unchanged. ReLUs are the recommended activation function for modern neural networks [6] we use these here as well. Since the linear parts of the ReLU and the linear matrix multiplication are differentiable we can employ gradient descent to train our classifier.

4.2. Support Vector Machines

Support Vector Machines (SVM) attempt to separate data along hyperplanes [1]. SVM offer a convex alternative to MLPs [21]. A convex classifier is guaranteed to converge to a global solution regardless of its initialization. Non-separable data can become separable if projected to a different possibly high dimensional space through a Kernel function. The Kernel trick allows SVM to solve some non-linear classification problems [21]. Radial basis functions (RBF) are commonly chosen as kernels [21], therefore, we choose to follow this practice as well.

4.3. Decision Tree

Decision trees are essentially binary trees, which work with the input features at the roots, decisions are made by moving up the tree to top leaves. At every branch in the tree, a decision is made, until one arrives at the decision [13]. For numerical input features, each decision partitions the data [1]. For example by comparison to a threshold value. To construct the tree these cutoff values have to be chosen at every branch. The construction problem turns into an optimization problem when a split criterion is introduced. Common choices are cross-entropy of Gini-coefficient [1]. During construction, the chosen function must be minimized.

Data of interest	Server Source	Variable
Conveyor speed	Siemens PLC	Conveyor: Gate Conveyor: QC Conveyor: Store
UR3 Position	Siemens PLC	*xyz* Gripper position
Action Flags	Siemens PLC	UR3 gripper open/close Grab/drop disk based on disk type
Quality control	SICK SIM4000	OK-NOK label *xy* disk deviation from the center point Absolute disk deviation
Gate position	SICK SIM4000	Position Speed Safe to go
Store	SICK SIM4000	Disk Size

Table 1: Industriy 4.0 demonstration cell: Relevant data linked with its reference source server.

Use Case	Description
Use Case 1	Conveyor speed: Slow Robot position: OK
Use Case 2	Conveyor speed: Slow Robot position: NOK
Use Case 3	Conveyor speed: Fast Robot position: OK
Use Case 4	Conveyor speed: Fast Robot position: NOK
Use Case 5	Conveyor speed: Too Fast Robot position: OK
Use Case 6	Conveyor speed: Too Fast Robot position: NOK

Table 2: Industrial 4.0 demostration cell: Controlled Use-Cases. Use-Cases which the value *Robot position: NOK* and *Conveyor speed: Too Fast* will result in NOK pieces.

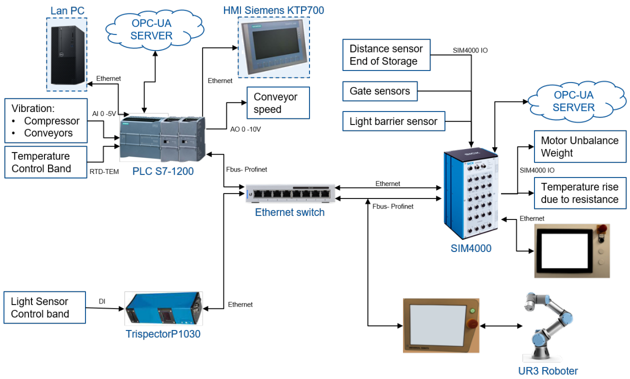

Fig. 2: Industry 4.0 demostration cell: Connection diagram (simplified).

5. Experiments

5.1. Data acquisition

Industrial projects require to work with a variety of sensors and controllers. Thus, there is a risk of incompatibility between technologies. To avoid that problem, universal protocols are often implemented at industrial projects. This is the case for our industrial 4.0 demonstration cell, which shares data via the protocol OPC-UA.

Figure 3 describes a simplified version of the real connection diagram of the demonstration cell. For our case, the data is acquired independently from two OPC-UA servers. This represents the interaction between multiple servers in real production environments.

In our implementation, we acquired the data in two steps. The first step was list all the variables of interest and to map their source. The second step was to determine the use cases of interest. Our study focussed on the data described in Table. 1 and the six use cases described in Table. 2. The data was manually collected by using the software ”UAExpert - v1.5.1”. Approximately 15 hours of data were collected, while equally sampling each use case during the data acquisition time.

5.2. Data preprocessing

Once enough data is collected, the next step is to clean the raw data and organize it accordingly [6]. For our case, that means to merge the data collected from the two OPC-UA servers, and synchronize the time-stamp. Since our systems are not completely synchronized, it was needed to calculate the delta-time between servers by referencing each server's base clock. Once the delta-time was determined, a time-stamp function was implemented to correct the time-shift between samples and be able to merge all the data without incompatibilities.

Data tags interpretation is an important step in the preprocessing process [6]. The raw data collected from the sensors is usually tagged by an alphanumeric code predefined by the sensor manufacture. To work with the data, it was needed to implement diverse functions which have the task to clean the raw data in a more human-readable form. This means to remove samples out of clear boundaries and missing values, as well as to translate the sensor code tags based on a dictionary predefined by the authors. This last step, might be optional in other implementations, however, it proves to be a great addition for testing purposes.

Once the data is clean and compiled in one directory, our step was to split the data based on the assembly cycle. For our purpose, we split the dataset into individual data-samples with help of the action tags included in the dataset Table. 1. Subsequently, each data-sample is automatically checked for missing sensor values and discarded if so. At this point, we have a se-

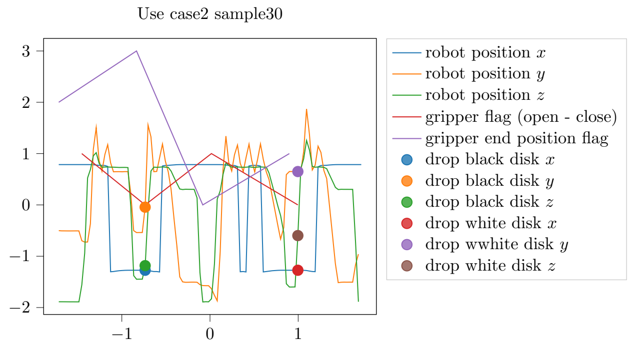

Fig. 3: Robot arm movement patterns. We show the movement of the cell's robot arm, its grippers, and the encoding of the overall position in the system. The changes in the grip flag are triggered, when the gripper opens or closes. Zero indicates an open, while one indicates a closed gripper. The position flags marks the state of the cell's arm. 0 means the black disk is in store, one that the black disk is at the gate, two that the white disk is in store and three that the white disk is located at the gate.

ries of data-samples based on time series. However, we are not interested in a time series analysis since we want to focus on ML methods based on an atemporal vector approach, therefore it was needed to compress each time-series based data-sample in an atemporal data-sample. To eliminate time as a variable in our dataset, we defined a series of rules to be applied in each assembly cycle data-sample:

- Value of the robot position (xyz) while dropping the disk in the storage area
- Mean conveyor speed value per conveyor band
- Absolute deviation (xy) value at the quality control check
- Absolute quality control value: OK - NOK

Each time-series sample was evaluated based on the rules stated above, and the result of each evaluation was considered as our atemporal data-sample, which we used as the base for our classifier.

5.3. *Classifier optimization and testing*

The robot's position at disk drop is marked in Figure 3. The quality prediction problem is framed as a classification task. The input vectors which we feed into our classifiers consist of the arm positions at the disk drops for both disks in three dimensions, as well as the largest recorded belt speed of all three belts. The interpretation of the quality measurement is used as the training target. It can be OK or NOK which we encode as zero and one.

We work with a total of 528 measurements. Each containing arm and belt data logs of individual cell run. 50 samples are set aside at random for testing purposes, leaving 478 training samples. The random number generator seed is set to one to ensure the train and test set splits are identical for all experiments.

We compare a total of three different classifier architectures on the data. A Support Vector Machine (SVM), a multilinear

Approach	accuracy
Naïve-Baseline	64 %
Support Vector Machine	82 %
Multilayer Perceptron	88 %
Decision Tree	98 %

Table 3: Comparison of Support Vector Machine (SVM), Multilinear Perceptron and Random Forest classification on our anomaly detection task. The first row shows the performance of naively predicting a broken piece every time.

		True Values		
		OK	NOK	Total
Predicted Values	OK	18	0	18 + 0 = 18
	NOK	8	24	8 + 24 = 32
	Total	18 + 8 = 26	0 + 24 = 24	50

(a) Support Vector Machine.

		True Values		
		OK	NOK	Total
Predicted Values	OK	18	0	18 + 0 = 18
	NOK	6	26	6 + 26 = 32
	Total	18 + 6 = 24	0 + 26 = 26	50

(b) Multilayer Perceptron.

		True Values		
		OK	NOK	Total
Predicted Values	OK	17	1	17 + 1 = 18
	NOK	0	32	0 + 32 = 32
	Total	17 + 0 = 17	1 + 32 = 33	50

(c) Decision Tree.

Table 4: Confusion matrices result of the prediction of 50 test data-points. We present the result of the predictions given by all three algorithms in detail based on the comparison between true values vs and predictions. The tables read from the upper left cell as: true-positive, false-positive, false-negative, true-negative.

perceptron (MLP) and a Random Forest structure. All three methods are trained using standard hyperparameters. [1]

Results are shown in table 3. For 32 of the total 50 test samples quality measurements indicate a problem. This sets the baseline over the entire data set, which would be obtained by simply labeling all samples as faulty. Over the test set, we require to classify more than 64% of the data correctly. We would therefore expect any naive classifier to produce at least 64% accuracy, which we have to surpass. In Table 3, we observe that this is indeed the case for all three approaches evaluated here.

Based on the MLP prediction results presented on Table. b shows that even though the prediction model is not 100% accurate, the percentage of false-positive predictions is 0%. The false-negative results represent the 12% of the test sample and the true predictions constitute 88% of the test data.

The outcome of the models shows that the random forest performed best followed by the multilinear perceptron and the support vector machine. However, by reviewing the confusion matrices presented in Table, 4, multilinear perceptron would be the preferred solution to be implemented at industrial applica-

[1] To allow exact reproduction of these results source code is available at `https://github.com/manubrain/Demo-Cell-Classification`

tions due to the 0% ratio in false-positive results and high accuracy of 88%.

6. Conclusion

Based on the explorative literature review, we observed that the hundred percent quality check method proves to be the most efficient method to guarantee the final product quality, however, the high implementation cost is its biggest draw-back. On the other hand, we have the statistical methods, which are known to be fast and efficient, but the lack of reviewing each product until the next bach review might imply a considerable loss of the produced products whenever NOK products are found. This is based on the uncertainty of when did the error occurs and how many products are affected.

By implementing ML models capable to predict the quality of the product, based on just the initial conditions of the process would be a great asset for the industrial processes. Within the limits of our work, we predict with 98% certainty the product quality by implementing a decision tree model with a 6-variables vector as input. However, our decision for industrial applications would be the MLP with 88% accuracy as to guarantee that all passed parts are OK and avoid labeling a NOK piece as OK. For further work, we expect to increase the accuracy of the prediction model by collecting more data-samples and experimenting with other, more complex, ML models.

It is important to state that this prediction model is not yet ready to fully replace the statistical methods, not the hundred percent quality check method, since this research is still on its early stages and need further development. However, a combination of SPC along prediction models could work well together, since the prediction model will evaluate each produced piece, and this might compensate one disadvantage of the statistical methods.

It is worth to mention the most important points learned during the project implementation. This might help as building blocks for further research in the area.

6.1. Lesson Learned

Incompatibility between technologies; devices configured on the same network presented troubles to share data within each other due to network restrictions, as well as the limitation to access and modify restricted program-code given by the machine manufacture. To overcome the situation stated above, it was necessary to reprogram big sections of the machine control system.

Data synchronization; two OPC-UA servers were used on this implementation. This lead to a series of data synchronization problems due to differences on the internal clock of both servers. In order to solve the issue, it was needed to write a time-stamp synchronization script that was deployed during the data preprocessing phase. Without harmonious time-stamps, determining the production cycle would have been impossible and the input data for the ML model would have been useless, and the model inaccurate.

Data interpretation; this process is considered complicate, time-consuming, and fundamental step on any ML applications. It is fundamental to understand the assembly process, the timing from the machine movements, the boundaries of the machine's sensors, as well a basic correlation within the presented data.

Missing values; to minimize the risk of false results delivered by the ML model, it would be compulsory to carefully review the ML input data. Missing values can have a great impact on the ML prediction, that is the reason why we deliberately review all the input data and validate it with the expected output from the diverse sensors. Whenever there was missing data, and as long as we did not fabricate data, we employed statistical methods to fill the data gap.

Adapting the industry 4.0 demonstration cell; the initial state of the demonstrator was very limited. Even though the process was well known, the lack of labels or flags whenever certain actions occurs, made complicated to have accurate data sampling from each assembly cycle. In order to overcome this uncertainty, we modify the machine program to raise specify flags whenever certain actions would be executed.

Machine Learning; careful measurement and meaningful preprocessing of our cell-date was key to the successful application of all three algorithms. Without the labels, the problem could not have been framed as a supervised classification problem.

Dataset size; as the starting point we collected as many data points as the ones found in the iris Dataset [5]. The next step was to collect batches of data of around 30 minutes per use case. This process was repeated until accumulate around 15 hours of data.

Acknowledgments

This work was funded by the European Regional Development Fund (ERDF) with the project name "ManuBrain"

References

[1] Aggarwal, C.C., 2014. Data classification.
[2] Aumi, S., Corbett, B., P., M., 2012. Model predictive quality control of batch processes. American Control Conference .
[3] Bishop, C.M., 2006. Pattern recognition and machine learning. springer.
[4] Delen, D., Demirkan, H., 2013. Data, information and analytics as services. Decision Support Systems 55.
[5] Fisher, R.A., 1936. The use of multiple measurements in taxonomic problems. Annals of Eugenics 7, 179–188. doi:10.1111/j.1469-1809.1936.tb02137.x.
[6] Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y., 2016. Deep learning. volume 1. MIT press Cambridge.
[7] Hinckley, C.M., 1997. Defining the best quality-control systems by design and inspection. Clinical Chemistry 43, 873–879. URL: https://doi.org/10.1093/clinchem/43.5.873, doi:10.1093/clinchem/43.5.873.
[8] Illés, B., Tamás, P., Dobos, P., Skapinyecz, R., 2017. New challenges for quality assurance of manufacturing processes in industry 4.0. Solid State Phenomena 261.
[9] Kahle, W., Liebscher, E., 2013. Zuverlässigkeitsanalyse und Qualitätssicherung. Oldenbourg Verlag München.
[10] Krauß, J., Dorißen, J., Mende, H., Frye, M., Schmitt, R., 2019. Machine learning and artificial intelligence in production: Application areas and

publicly available data sets. Production at the leading edge of technology 9.
[11] Krauß, J., Pacheco, B.M., Zang, H.M., Schmitt, R., 2020. Automated machine learning for predictive quality in production. Procedia CIRP 93.
[12] Kurniati, N., Yeh, R.H., Lin, J.J., 2015. Quality inspection and maintenance: the framework of interaction. Procedia Manufacturing 4.
[13] Marsland, S., 2015. Machine learning: An algorithmic perspective.
[14] Mitra, A., 2016. Fundamentals of Quality Control and Improvement. wiley.
[15] Ritter, C., Schweitzer, F., 1992. Neue wegene der prozeßmodellierung für die rechnergestützte qualitätsvorhersage und -sicherung in der spanplattenfertigung. Holz als Roh- und Werkstoff 50.
[16] Robinson, H., 2011. Using poka-yoke techniques for early defect detection.
[17] Sandoval-Chávez, D.A., Beruvides, M.G., 1998. Using opportunity costs to determine the cost of quality: A case study in a continuous-process industry. The Engineering Economist 43, 107–124. doi:10.1080/00137919808903192.
[18] Selmaier, A., Robitzch, P., Mayr, A., Fürst, J., Franke, J., 2019. Cross-process quality analysis of x-ray tubes for medical applications using machine learning techniques, in: Wulfsberg, J.P., Hintze, W., Behrens, B.A. (Eds.), Production at the leading edge of technology, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 513–522.
[19] Selvamuthu, D., Das, D., 2018. Introduction to Statistical Mehods, Design of Experiments and Statistical Quality Control. Springer Nature Singapore Pte Ltd.
[20] Sohnius, F., Schlegel, P., Ellerich, M., Schmitt, R.H., 2019. Data-driven prediction of surface quality in fused deposition modeling using machine learning, in: Wulfsberg, J.P., Hintze, W., Behrens, B.A. (Eds.), Production at the leading edge of technology, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 473–481.
[21] Suykens, J.A., Van Gestel, T., De Brabanter, J., De Moor, B., Vandewalle, J.P., 2002. Least squares support vector machines. World scientific.