Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

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.

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

1. Introduction

Product features such as reliability, durability, or complete functionality determine to a large extent the success and profit of companies on the market. That is why it is highly important for companies to ensure that every product meet the expected requirements. Throughout the years, diverse methods for quality control (QC) have been presented and implemented in production. Some of those methods prove to be highly effective, but costly, i.e. the revision of every produced piece, either by automated or manual methods, or by doing quality inspections based on a portion of the produced pieces and extend the results based on statistical methods. Which still has a considerable risk of not detecting NOK parts and sending them to costumers [14, 6, 9].

Whenever a QC method is selected, it is important to remark that compromises must be made in most cases. For example, traditional methods like controlling every produced piece might affect the production goals by delaying the production speed at the cost of lower rejection rates On the other hand, higher production speeds normally requires statistical quality control methods. However, those methods do not provide one hundred percent certainty that the required product quality will be achieved [16, 12].

With the evolution of machine learning (ML) applications, we expect to enhance the QC by predicting the product quality at the initial stage of production. This will not only reduce cost in production, and strengthen the QC process, it will also be part of the predictive analytics tools found in production, which is currently widely used in diverse fields due to its accurate predictions [2, 10].

This paper is divided in 6 sections:

- **Introduction**, states the motivation of the project as well as the problem it intended to be addressed.
- **Related Works**, covers the methods used in quality check in production, as well as the bases of predictive analytics in production.
- **The case industry 4.0 demonstration cell**, describe the demonstration cell used on this project. It answer the

question of where to test predictive analytics in a controlled environment and why our system is a valid platform for this implementation.

- **Machine learning Methods**, explains the three machine learning methods implemented on this work: Multilayer perceptrons, Support Vector Machines, Decision Tree.
- **Experiments**, describe the experiment process from the data acquisition and preprocessing, to the classifier optimization and testing.
- **Conclusion**, express 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.

2. Related Works

2.1. *Quality check in production*

The quality control 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 [8]. According to Illés, there are three established tools and methods of quality control [8]: The Statistical Process Control (SPC), Auditing and Total Quality Control (TQC). The SPC 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 [16]. SPC methods can also take place before the actual manufacturing process, this is commonly known as "off-line SPC".

During the design phase of a process or product, off-line SPC procedures are deployed. Those procedures aim to increase the quality of the outcoming product by choosing controllable products and process parameter [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].

Another conservative technique to perform the quality check would be to check 100% of the produced parts. On this method, every product is controlled and binary sorted as an OK or Non-OK part. However, this has the disadvantage of high cost related to personal or specialized material and in certain cases, could lead to a delay in production.

Product and process auditing is the independent examination of product and process quality to provide information and it is considered the most fundamental quality control technique. This method is executed by an independent auditor, who based on sampling or checking records limits the finding on reporting incorrect processes procedures or products documentation. In a further step, the findings would be later corrected by another person rather than the auditor [6].

Statistical methods such as acceptance sampling or quality control charts represent a compromise between zero and the hundred percent control [16, 12].

While the SPC mostly optimizes the capability of the process, for example the acceptance plan prevents the nonconforming product to be pass and then delivered to the next process or the customer [12]. In the case of a hundred percent control, this is associated with higher costs and a higher requirement of time, compared to a quality control based on a machine learning model. Generally, statistical methods are faster than the hundred percent controls, however, their disadvantage is that those methods are based on assumptions. those assumptions might lead to errors if the variance of the product quality higher than expected.

The concept of TQC is to extend the scope of quality control to the whole company and product life cycle, by involving all departments of the organization [8]. The quality characteristics of a product can be classified as variables or attributes. Variables are measurable characteristics, which are shown as numbers. On the contrary, attributes are not measurable and can only take the binary values like "go" or "no-go" [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. This allows the desired quality to be restored. With that in mind, predictive quality control is a promising solution.

2.2. *Predictive analytics in production*

According to Delen and Damirkan, predictive analytics (PA) consist on unravel 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 there is enough data available. By considering the industrial and production environment, PA would be a great asset due to the high flexibility and high amount of data available coming from processes, single machines or cells, products, and others. Some common industrial applications of PA at the industry are focused products (design and optimization), machines (predictive maintenance) and production (scheduling) [10].

Applications that combines PA with QC might be a great improvement in the quality control process because it is an improvement compared with the earlier mentioned methods 2.1, in terms of speed, accuracy and time delay.

The outcome of an explorative literature review on this matter, shows that the findings that combines PA and QC are very scarce in the literature. This could be in great part, due to the novelty of the area [1, 15]. Related work, e.g from Krauß focused on automated machine learning for predictive quality in production [11] and a product quality prediction in a process chain [10].

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 on this area. Related works, e.g from

Krauß focused on automated machine learning for predictive quality in production [11]; product quality prediction in a process chain [10]; model development for predictive quality control of batch processes [1]; and ways of process modeling on quality prediction and assurance of chipboard [15]. Considering the lack of literature, our goal is to decrease this gap on the research area by implementing a functional predictive model in quality assurance for an assembly process, as well as describing the lesson learned during this implementation.

3. The case industry 4.0 demonstration cell

To test our theories, we need to select the right testing platform. This platform should represent real industrial applications and be able to handle a certain level of complexity. In our work, we applied a case-based research approach [18] in order to answer the research question. One central motivation for a case-based research approach is to gain insights for real needs, rather than to develop theories without practical relevance [3].

Our case of study is an abstraction of an assembly process at the industry. The industry 4.0 demonstration cell is composed by three independent conveyor belts, a robotic assembly arm, a laser scanner used for quality control as well as a wide range of sensors, all orchestrated by a SIEMENS PLC S7-1200. The assembly process done at this demonstration cell consist on stacking two disks of different size on the top of each other. The quality of the pseudo products is later evaluated by the laser scanner by evaluating the concentricity of both disks Figure 1. If the disk's concentricity is within a tolerance of 1.5mm, the piece is classified as OK, if not, it shall be classified as NOK. By considering that malfunctions can occur during real production processes at industries, our demonstration cell is able to simulate failures in diverse areas, such as:

- Assemble errors due to the robotic arm
- Bearing damage on the conveyor belt
- Resistance on the conveyor belt
- Leakage in the compressed air system
- Productivity limit reached
- Missing material

Those simulated failures have a real 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 parameter 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. The findings gained with the demonstration cell can be transferred to various real assembly processes and use cases, due to its manageable size and a high degree of scalability. This would be a step forwards for future implementations at the industry, where the quality predictions might determine early enough, the next steps for the assembled pieces.

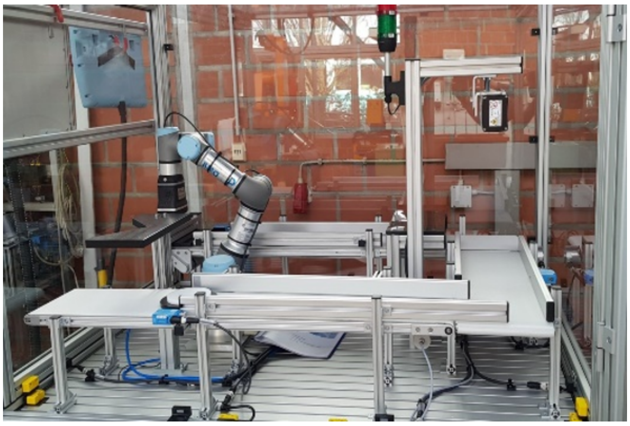

Fig. 1: The industry 4.0 demonstration cell at the University of Siegen

4. Machine Learning Methods

We compare Support Vector Machines (SVM), Multilinear Perceptrons [2] as well as random forests in 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 [2]. 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 [7] 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) offer a convex alternative to MLPs [17]. A convex classifier is guaranteed to converge to a global solution regardless of its initialization. The Kernel trick allows SVM to solve non-linear classification problems [17] by mapping implicitly mapping the inputs into a higher feature space. Radial basis functions (RBF) are commonly chosen as kernels [17] we choose to follow this practice as well.

4.3. *Decision Tree*

Decision trees are binary trees, these binary trees work with the input features at the roots, resulting decisions are found by moving trough the tree to top leaves. At every branch in the tree a decision is made, until one arrives at the final conclusion

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 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 with Robot position: NOK and Conveyor speed: Too Fast will result in NOK pieces.

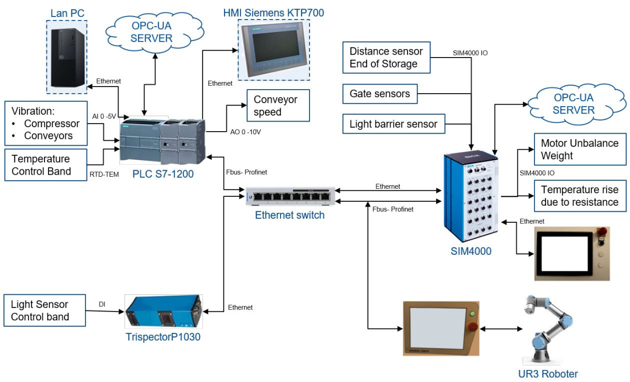

Fig. 2: Industry 4.0 demostration cell: Connection diagram (simplified)

[13]. Decision trees are trained using **(MW: Will write more tomorrow.)**

5. Experiments

5.1. Data acquisition

Industrial projects requires to work with a variety of sensors and controllers. Thus, it might occurs incompatibility between technologies. In order 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 protocol OPC-UA.

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

The first step to collect the data is to list all the variables of interest and to map their source. The second step is to determine the use cases of interest. On this research, we focussed on the data described on Table. 1 and the six use cases described on Table. 2. The data was manually collected by using the software "UAExperts". Approximately 15 hours of data was collected, by 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. For our case, that means to merge the data collected from the two OPC-UA servers, synchronize the time stamps. Since our systems are not completely synchronized, it was needed to calculate the delta-time based between servers based on the base clock from each server. 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. The raw data collected from the sensors is sometimes tagged by an alphanumeric code. In order to work with the data, it was needed to implement diverse functions wich 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 sensors code based in a predefined dictionary settle by the authors. This last step, could 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, the next step would be to split the data based on the assembly cycle. For our purposed, we split the dataset in 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

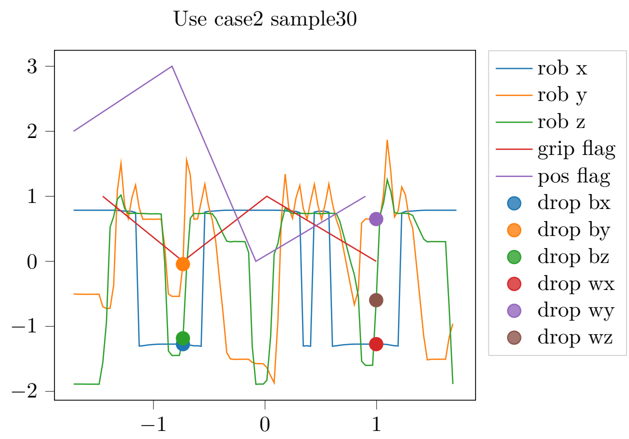

Fig. 3: Robot arm movement patterns. We show the movement of the cell's robot arm, its grippers, and the encoding of the overall position in the system. The changes in the grip flag are triggered, when the gripper opens or closes. Zero indicates an open, while one indicates a closed gripper. The position 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.

have a series de data-samples, which each of them is based in time series. However, we are not interested in a time series analysis, therefore it would be 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:

- Robot position (*xyz*) while dropping the disk in the storage area
- Mean conveyor speed per conveyor band
- Absolute deviation (*xy*) at the quality control check
- Absolute quality check result: 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 would be used by 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 consists 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 good or bad 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 runs. 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 perceptron (MLP) and a Random Forest structure. All three methods are trained using standard hyperparameters. [1]

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.

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

(a) SVM.

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

(b) MLP.

Predicted Values		True Values: OK	True Values: NOK	Total
	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 matrix result of the prediction of 50 data-points.

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 beat. 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 represents 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 on Table, 4, multilinear perceptron would be the option of choice to be implemented at industrial applications due to the 0% ratio in false-positive results.

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

6. Conclusion

Based on the 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 individual product until the next bach review might imply a considerable loss whenever NOK products are found. This based in 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, just by the initial conditions of the process would be a great asset for the industrial processes. Within the limits of our work, we are able to 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 the QC and avoid labeling a NOK piece as OK. For further work, we expect to increase the certainty 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. However, a combination of SPC along prediction models could work well together, since the prediction model will evaluate each produced piece, and this will somehow compensate one of the disadvantages from 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, the input data for the ML model would have been useless and the model inaccurate.

Data interpretation; complex, 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 of a basic correlation within the presented data.

Missing values; in order 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 in 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 project was partly supported by the German and European IKT.NRW project "'ManuBrain"'

References

[1] S. Aumi, B. Corbett, and M. P. Model predictive quality control of batch processes. *American Control Conference*, (5646-5651), 2012.

[2] C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006.

[3] D. Cutcheon and J. Meredith. Conducting case study research in operations management. *Journal of Operations Management*, 11(3):239–256, 1993.

[4] D. Delen and H. Demirkan. Data, information and analytics as services. *Decision Support Systems*, 55(359-363), 2013.

[5] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

[6] M. Fox. *Quality Assurance Management*. Springer, 1993.

[7] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[8] B. Illés, P. Tamás, P. Dobos, and R. Skapinyecz. New challenges for quality assurance of manufacturing processes in industry 4.0. *Solid State Phenomena*, 261(481-481), 2017.

[9] W. Kahle and E. Liebscher. *Zuverlässigkeitsanalyse und Qualitätssicherung*. Oldenbourg Verlag München, 2013.

[10] J. Krauß, J. Dorißen, H. Mende, M. Frye, and R. Schmitt. Machine learning and artificial intelligence in production: Application areas and publicly available data sets. *Production at the leading edge of technology*, 9(493-501), 2019.

[11] J. Krauß, B. M. Pacheco, H. M. Zang, and R. Schmitt. Automated machine learning for predictive quality in production. *Procedia CIRP*, 93(443-448), 2020.

[12] N. kurniati, R. H. Yeh, and J.-J. Lin. Quality inspection and maintenance: the framework of interaction. *Procedia Manufacturing*, 4(244-251), 2015.

[13] S. Marsland. Machine learning: An algorithmic perspective, 2015.

[14] A. Mitra. *Fundamentals of Quality Control and Improvement*. wiley, 2016.

[15] C. Ritter and F. Schweitzer. Neue wegene der prozeßmodellierung für die rechnergestützte qualitätsvorhersage und -sicherung in der spanplattenfertigung. *Holz als Roh- und Werkstoff*, 50(62-66), 1992.

[16] D. Selvamuthu and D. Das. *Introduction to Statistical Mehods, Design of Experiments and Statistical Quality Control*. Springer Nature Singapore Pte Ltd., 2018.

[17] J. A. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. P. Vandewalle. *Least squares support vector machines*. World scientific, 2002.

[18] R. Yin. *Case study research and applications: Design and methods*. Sage publications, 2017.