Closing the loop: Real-time Error Detection and Correction in automotive production using Edge-/Cloud-Architecture and a CNN

Johannes Vater[†][§], Maximililian Kirschning[†] and Alois Knoll[§]
[†]Planning and Production of Electrified Powertrains, BMW Group, Munich, Germany
[§]Chair of Robotics, Artificial Intelligence and Real-time Systems, Technical University Munich, Munich, Germany
Email: Johannes.JV.Vater@bmw.de, Maximilian.Kirschning@rwth-aachen.de, knoll@in.tum.de

Abstract—— **One challenge faced by the automotive industry is the shift from combustion to electrically powered vehicles. This change strongly impacts on components such as the electric motor and the battery, and hence on production. In this context, the low level of expert knowledge is especially problematic. To meet these new challenges, this paper introduces a data-driven optimization of the production process by integrating a modular edge and cloud computing layer, and advanced data analysis. Defects are classified by a convolutional neural network (CNN) (predictive analytics) and corrected (depending on the defect type) by an automated rework (prescriptive analytics). The architecture of the CNN achieves an accuracy of 99.21% to predict the defect class. The automated rework process is selected through an implemented decision tree. The edge device communicates with a programmable logic controller (PLC) through a cyber physical interface. As an example of its practical application, the method is applied to hairpin welding of the stator of an electric motor with real production data.**

Index Terms—**industry 4.0, cloud computing, edge computing, machine learning, convolutional neural networks, electric motors, hairpin, predictive analytics, prescriptive analytics, prescriptive automation**

I. Introduction

Early detection of quality deviations is very important but poses a major challenge in the manufacturing industry [1]. Modern production processes can be continuously monitored with the help of industrial Internet of Things (IIoT) networks. However, traditional and manual quality monitoring using IIoT networks is not easily accessible, and is not possible in real-time [2]. Therefore, real-time and automated fault detection without human intervention is essential for IIoT networks [1]. In addition to these networks, artificial intelligence offers great potential for error detection and classification. The advantage is that dependencies and quality deviations are not modelled based on expert knowledge, but dependencies based on data are learned automatically by the classification algorithm. Such objective data-based learning is highly advantageous for the electrical powertrain production of newly developed battery electric vehicles. Unlike combustion engine production, almost no expert knowledge on electric vehicle production is available and new technologies are needed [3]. A new technology called hairpins. Hairpins are copper rods that replace the traditional copper windings of a stator. This can increase the efficiency of an electric motor. However, the pair of hairpins must be welded for electrical connection (see Section III-A). However, this welding process is unstable and susceptible to faults. Any welding defects must therefore be detected and classified, and then automatically corrected.

For this purpose, we construct a CNN that automatically detects and classifies weld defects in the production line. Based on our previous research, we compare the three-dimensional (3D) data with the black and white (BW) images as an input to the CNN. To better compare the input data, we summarize the results of our previously published paper [4]. As explained in Section III, our CNN modeling is analogous to the cross-industry standard process for data mining.

We next propose an IIoT network for real-time processing the 3D scans or BW images by the developed CNN. As the input images have a large file size, their real-time processing requires computationally intensive hardware. This hardware must be capable of receiving and processing the data without high latency, and must adjust the production line without human intervention. Edge computing provides the opportunity to localize functionalities nearby by connecting directly to the production line [5]. Our proposed architecture (described in Section IV) combines edge- and cloud-computing, enabling automatic defect detection by the CNN. Depending on the predicted defect case, the edge computing device performs an automated rework without human intervention. By virtue of its modular design, the solution can also be integrated into other use cases.

II. State-of-the-Art

A. Optical detection of quality deviations in automotive production by CNNs

In the field of automobile production, there are currently only a few approaches for the detection of quality deviations with the help of a CNN. Examples are the detection of cavities in blow holes and in the production of wheel rims or steering knuckles [6]. In addition, other publications deal with the detection and classification of welding spots on a car door [7] or the detection of scratches in car paint [8]. Within the production of electric drives, the authors present a CNN to check the ultrasonic crimp connection of a stator, because in addition to the assembly of the winding, the process of contacting the wire ends with cable lugs offers great potential

978-1-7281-6371-0/20/$31.00 ©2020 IEEE

for optimization [9]. In addition, Mayr et al. dealt with the detection of welding defects in hairpin welding [10].

None of the authors listed above achieved an error detection accuracy greater than 99%. However, higher detection accuracy is essential for the successful implementation of serial quality monitoring; otherwise, components that fail the quality requirements will leave the factory, reducing the reputation of the company and even endangering the safety of humans, animals, or the environment. The goal of the present work is thus to achieve an accuracy of at least 99%, which would be a significant improvement over the current state-of-the-art.

B. Edge computing in the production industry

This Section discusses the state-of-the-art for the application of edge- and cloud-computing in the manufacturing industry and in automotive production. The goal of such an architecture is the real-time detection of quality deviations (predictive analysis) and the generation of a recommendation for action (prescriptive analytics). Since the authors have already published a good overview of the approaches, we refer to their paper for a closer look [11]. In summary, however, they figured out that no publication exists whose architecture is suitable for short production cycle times. Furthermore, these edge- and cloud-architectures do not have a modular structure and cannot be managed centrally. As a result, Ref. [11] describes and models such an architecture that solves the deficits. However, the recommendation for action is given to an employee who must manually intervene in the process. Therefore, the second goal of the architecture published in this paper is to extend the architecture of Ref. [11], in order to automatically execute these recommendations through a closed loop feedback system.

III. Predictive Analytics

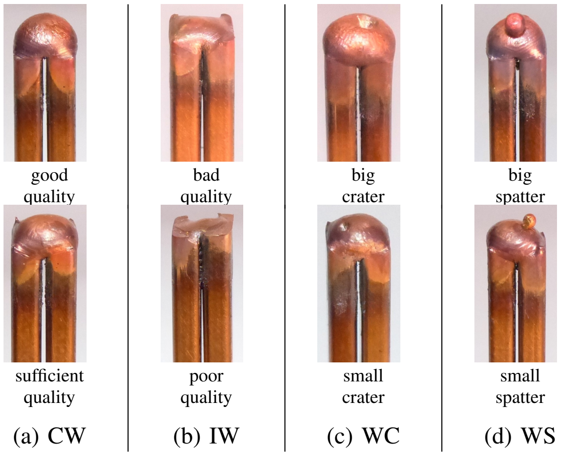
A. Business Understanding

An important production step in stator manufacturing is the electrical and mechanical connection of the hairpins. The pins are joined by automated laser welding [12].

The difficulty is that the automated welding of copper leads to different welding faults. A major disadvantage is that copper absorbs less than 5% of laser irradiation at near $1000nm$ wavelengths. To overcome the strong reflective properties, the laser beam welding requires a very intense power density [13]. However, as shown in Fig. 1, this leads to irregular weld seams [14].

As the current production line lacks an automated error classification system, the stator enters all processing steps until the end-of-line test. A stator with a defective weld must first be removed from the production line. Next, the welding tool is disassembled, the defect is inspected visually, the stator is aligned manually, the welding tool is reassembled, the stator is reinstalled, and the defective welds are re-welded.

To improve the efficiency of this time-consuming and expensive process, the welding defects must be **detected** and **classified** in the welding station, enabling targeted and automated re-welding in large-series production. Moreover, the detection, classification and rework of the hairpins cannot exceed the cycle time ($30s$).



(a) CW (b) IW (c) WC (d) WS

Fig. 1: Representation of the four quality classes after welding pairs of hairpins.

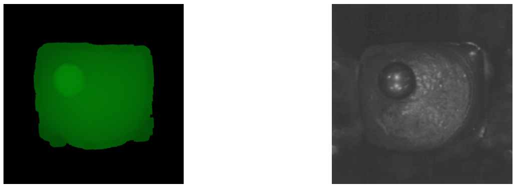
The defects presented in Fig. 1 were obtained after consulting a technologist in this field. When welding the hairpins during the series production of electric motors, the possible errors are insufficient welding (IW), welding craters (WC), and welding spatter (WS). For comparison, a correctly welded (CW) hairpin pair is also displayed in Fig. 1. In addition, the severity of the error has been divided into two further classes.

B. Data Generation

The quantity, quality and distribution of classes in the dataset strongly affect the training process of a CNN. Therefore, the data must be generated in a structured way.

Birds' eye views of the welded hairpins were recorded by a camera (XR-HT40M, Keyence) and digitized as 3D data showing the shape of the welding dome. The BW images were digitized similarly. The data-acquisition procedure collected approximately 560 BW pictures and 560 3D data of the welds in each class. Fig. 2 shows a 3D image and a BW image of the WS defect class.

As already explained, the quality of the CNN depends on various factors, including the quantity of the used data. Unfortunately, creating a large and balanced data set, involves a great deal of effort in production. To generate a sufficiently



(a) Exemplary 3D-scan of a hairpin with a WS (b) Exemplary black and white picture of a hairpin with a WS

Fig. 2: Recording of a hairpin welding with a weld spatter defect.

large dataset, the original images were duplicated by data augmentation procedure that generates additional synthetic images by combined rotation, displacement, and mirroring transformations.

It is important that these synthetically generated images are located exclusively in the training data set. For this reason, the generated original data was divided into 80% training data and 20% test data. Data augmentation was applied exclusively to the training data set. Using a factor of 49, the training data set could be expanded from approximately $1,800$ to $90,000$ images by mirroring, rotating and shifting. The resulting size of the training and test data set is listed in TABLE I.

C. Data Preprocessing

To maximize the quality of the CNN output, the utilized data must be pre-processed. Below we present the preprocessing steps of the 3D scans (the preprocessing steps of the BW images are very similar).

1) In the first step, the weld of the image is cut out in a size of $450x450$ around its center of gravity.
2) The 3D-scan of the Keyence camera is coded. Because of this, in the second step of preprocessing colour coding is converted into height information. The height is in a range from 0 to 16 mm.
3) Since only the dome of the hairpin weld is of interest, the following preprocessing step cuts the dome in a range of 6 mm in both positive and negative z direction around the median of the hairpin.
4) Due to the high volume of the generated images, caused by the serial production it is essential to compress the size of the image to reduce the memory requirements. This is done by compressing the weld to a size range of $30x30$ pixels in x and y direction. This reduction is achieved by average pooling. Furthermore, the height information is scaled into a range between 0 and 255.
5) In the fifth step of data preprocessing, the data is converted into a grayscale image.
6) Finally, the image is normalized based on subtracting the mean value of one pixel over the entire data set, dividing the standard deviation of these pixels by the standard deviation.

D. Architecture Modeling

The CNN is constructed from a series of convolution blocks. Each block consists of two successive filters including batch normalization (BN) and rectified linear unit (ReLu) activation, followed by a pooling layer. The number of filters doubles in each successive convolution block. The first layer uses eight filters of size $3x3$. This number doubles to 16 filters in the second block, 32 filters in the third block, and 2^{n+2} filters in the n^{th} block. Each block (except the last) is followed by a maximum pooling layer. The last block is followed by global average pooling rather than maximum pooling. This construct computes the one-dimensional average value of each multidimensional matrix of features resulting from cross-correlation with a filter. The one-dimensional feature vector contains 64 entries, one for each filter in the last convolution block (where the number of convolution blocks is four). The convolution blocks are followed by a fully networked layer with 32 neurons, a BN layer, a ReLU activation function, and a dropout layer with a dropout rate of 0.5. The final layer is an output layer with a neuron for each class and a softmax activation function, which outputs the probability of each class.

TABLE I: DIVISION OF THE DATASET

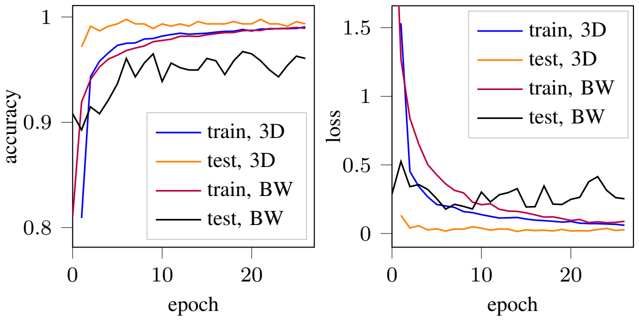
Class	Training set	Test set	Sum
IW	456	104	560
WS	455	102	557
WC	438	125	563
CW	478	126	604
Sum	1.827	457	2.284
augmented	91350	-	-

E. Evaluation

The proposed model was implemented in Keras (version 2.2.4) using tensorflow backend (version 1.14.0). We applied a stochastic gradient decent optimizer with a learning rate of $1e-4$, a decay rate of $1e-6$, and a Nesterov pulse of 0.9. In this classification task, the categorical cross entropy was selected as the loss function.

The architecture of the CNN defined in Section III-D was validated by five-fold cross-validation. Fig. 3 displays a typical training course of the five-fold cross-validation of the inputted 3D data and BW images. Panels (a) and (b) of this figure track the accuracies and losses, respectively, of the training and validation data over the training epochs.

Both courses show that the training process with the 3D data as input of the CNN is very stable with high accuracy and low loss. It should be emphasized that no fluctuations in the course can be recognized and furthermore, no over-or underfitting can be detected due to the regularization techniques built into the network. However, the BW images as an input show strong fluctuations.



(a) Resultant training and validation progress for model accuracy

(b) Resultant training and validation progress for model loss

Fig. 3: Training and validation results of the 3D input data (blue and orange, respectively), and of the BW input data (purple and black, respectively). Shown are the model accuracies (left) and losses (right) obtained by five-fold cross-validation.

TABLE II lists the average results of the five-fold cross-validation tests. The quality of the designed network architecture was further quantified by various evaluation metrics (precision, recall, and F_1-score).

The model with the 3D data input achieved an accuracy of 99.21%. Furthermore, the metric precision was 99.24%, confirming the goodness and reliability of the model results. The precision is an important metric in production processes, as it indicates whether or not faulty parts are falsely predicted as good parts that can be delivered to customers. This means that a weld which is in reality a defective weld, but which was predicted to be a good weld, is considered a false positive. The specified metrics precission and recall, are exclusively referred to the CW class. The recall and F_1-score also exceeded 99% in this case.

However, the BW images significantly degraded the model performance. When the BW images were input, the accuracy reduced to 95%, and the precision of the CW class was 98.49%, meaning that a bad weld was sometimes incorrectly classified as a good weld. Such misclassifications increase the chance of defective products being shipped to customers. Moreover, the recall was only 93.8%, meaning that a correct part was sometimes incorrectly classified as a bad part. A large number of these misclassifications increases the workload on the production line.

Although a simple camera is considerably cheaper than a 3D scanner, we recommend the latter for detecting and classifying welding defects, because defective welds delivered to customers will incur significant economic consequences, and will damage the company's reputation.

IV. Prescriptive Process Optimization

A. Mechatronic system

For a 100-percent inline inspection by a CNN and automated adjustment of process parameters an appropriate IT-architecture is needed. As an example, we designed a system for detecting defects in welded hairpin pairs, as described above.

The developed system combines mechanical, electronic, and information technology components. Fig. 4 shows the systematic structure of a mechatronic system. According to the VDI guideline 2206, a mechatronic system consists of four units forming a system-internal control loop [15]. The four units are described below.

- The **basic system** contains a mechanical, electromechanical, hydraulic or pneumatic structure, or a combination of these. The basic system (the welding station of the hairpins in this paper) is the core element of the physical layer.

TABLE II: Average Accuracy, Precission, Recall as well as F1 for all 5 runs

Input	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
3D-data	99.21	99.24	99.21	99.22
black-white image	95.0	98.49	93.8	96.1

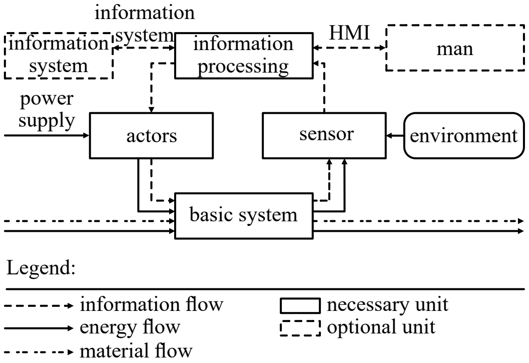


Fig. 4: Basic structrue of a mechatronic system.

- The **sensor** acquires the measured variables of the basic system, which are transmitted to the information system. The sensor unit is influenced by the environment. In this paper, the sensor that inspects the welding quality is a 3D scanner.
- **Information processing** is the central component of the logical level. This component determines the necessary actions based on the measured values of the sensors, and hence influences the state variables of the basic system.
- The **actor** directly influences the necessary action determined by the information system by adapting the state variables to the basic system.

These elements are connected by material, energy and information flows.

B. High-level design

In the present application, the basic structure of the mechatronic system was implemented by a edge- and cloud- architecture. The environment of the mechatronic system was divided into three sub-environments as shown in Fig. 5.

1) The production environment includes the basic system and its inputs and outputs. The products are processed and transformed. In our application, the production environment was the hairpin welding station.

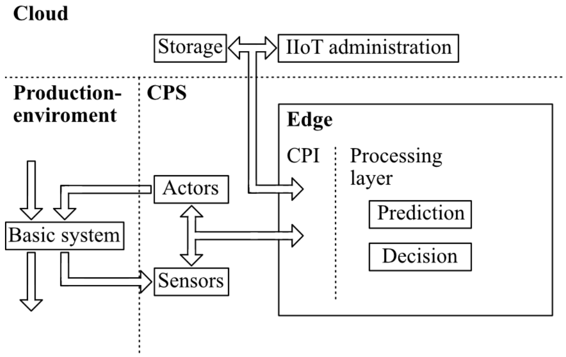


Fig. 5: Architecture of edge- and cloud-system.

2) The cyber physical system (CPS) provides the interface between the production level and the cloud. It contains the actuators, sensors and the information processing system. Its close proximity to the production line enables real-time data processing [16].
3) The cloud provides the data storage and organization platform. It coordinates the data collected by the multiple CPSs and stores them in a central location.

The main component of the CPS is the edge device. Located at the information processing system, it contains the necessary modules for the transformation and processing of the data collected by sensors and the cloud. It is administrated through the cloud by an IIoT administration platform. The modular internal structure divides the tasks into individual modules that can be processed in parallel. This also leads to the advantage of an easier change and provision of the individual modules, as the whole system never has to be replaced, but only individual modules. One potential disadvantage of the modular structure is the increased communication effort, but this disadvantage does not present in practice (as elaborated in Section IV-H). In the present application, the modules were implemented using the docker container platform. These modules are divided into two layers. First the cyper-physical-interface (CPI), which includes the communication modules for the production and cloud levels. And second the processing level, which handles the actual data processing. The two layers are illustrated in Fig. 5.

The overarching level is the cloud level, which (as described above) serves as the data storage and organization platform. Software that centrally manages multiple CPSs can manage, modify and monitor an unlimited number of edge devices, enabling quick deployability and therefore flexible allocation of production units. Central management also offers central monitoring capabilities, which can be supervised by a central production hub. As the cloud has immense storage capabilities, the data for the ongoing optimization of the CNN were stored here, as described in Section III-D.

The CPS and its internal structures in the given application will be discussed in subsequent sections.

C. Components of the Edge Device

The CPI is the central communication layer within the CPS (see Fig. 5). Besides connecting the information technology modules with the physical components, it enables communication between the external components such as cloud modules or other CPSs. The modules anchored at this level are exclusively intended for data forwarding. For example, one module passes communications to the storage providers as databases, and another module communicates with a PLC via Open Platform Communications United Architecture (OPC UA). The advantage of this approach is a clearly defined separation of communication and processing, which simplifies the interfaces and improves the maintainability.

These communication streams are then processed at the processing level. The incoming data are processed and changed by a single processing unit. The individual processing steps are implemented as modules that independently transform any data passed via defined inputs and outputs. The necessary communication between the modules is bilateral. Each module performs a specific task and forwards its results to the respective recipients. In this way, the tasks can be parallelized. During continuous production, the decision on how to proceed with a hairpin and the prediction of a subsequent hairpin can be made simultaneously. In the hairpin welding application, the modules were divided into prediction and decision-making modules.

D. Communication between Modules

The simultaneous processing and sending of data in different modules requires global communication between the respective modules. This communication differs from external communication, which sends bidirectional messages between the individual modules. Each module can push its data onto platforms called topics, and thereby distribute its data to any number of other modules. Moreover, via a subscribe mechanism, any module can listen to a specific topic and receive the data pushed onto it. This mechanism directs the data to its designated processing destination and avoids the additional overhead of interrogation communications. The message generation is standardized by the Message Queuing Telemetry Transport communication protocol (MQTT), which promises secure data transmission and fast communication. These can be confirmed by Section IV-H.

E. Prediction modules

The prediction modules are subdivided into preprocessing and prediction modules. The predictive-analytics modules convert the data to an error estimate associated with the analyzed stator. Note that this step of the error analysis provides an error estimate but does not recommend an action

1) Preprocessing: The preprocessing module operates by the procedure described in Section III-C. In the present application, several pins simultaneously recorded by the 3D camera were split and preprocessed individually. The image information was then pushed onto the prediction module via a topic.

2) Prediction: The prediction module contains the CNN presented in Section III-D. This module receives the image data from the preprocessing module and predicts one of the error classes presented in Fig. 1. The CNN is hosted within the module as a Representational State Transfer application program interface (API), which is separated from the framework that embeds it. Thus, updating the CNN through the cloud leads to an update of the prediction without having to change the actual module.

F. Decision modules

This group of decision modules forms the prescriptive part of the analysis. The modules are divided into a decision tree that determines the type of rework, and another decision tree that checks the feasibility.

1) Decision module: As shown in Fig. 6, the selected rework strategy depends on the error class and the probability of the prediction. First, the reliability of the CNN is checked. If the prediction $probability$ (pb) is below 80%, the component must be checked by an employee. The second step determines the type of rework based on the predicted defect class. If no defect is present (i.e., the class is CW), the component is forwarded to the next production step. If an error is present, it must first be checked whether repeat tests of the stator yielded consistently negative results (i.e., whether $rework(rw) > 1$). If true, no further rework is allowed. In the initial classification, a WC or WS fault is reworked under a laser intensity of 70%, whereas an IW is reworked under full laser power. The obtained decision is conveyed to the next module, which then checks the feasibility of the separation.

2) Validation module: The feasibility of a task depends on many and various factors, and especially on the cycle time. The rework time of the production system must not exceed its allocated time slot. If the rework time is too long, the component must be rejected. To distinguish such cases, timestamps are attached to each pin, allowing a precise observation of the cycle time (the timestamps will be detailed in Section IV-H). Another important factor is the status of the production environment. The post-processing strategy is applied only when the specified task is permitted within the safety or environmental constraints. Otherwise, the higher-level production system is notified and an emergency strategy is implemented. Thus, the feasibility module performs the final checking that prevents malfunctioning of the basic system.

G. Closing the loop

The validation module receives its data and send its results via a communication interface with a PLC. The PLC controls the processes in the basic system and contains the necessary routines of the rework concepts. To acquire information on the PLC status and the environment, the validation module requires bidirectional communication. This is achieved by a communication module located in the CPI. Assisted by an OPC UA server created on the PLC, the communication module can subscribe to the validation module, request the status information, and set control bits. The status information is then passed to the validation module for processing and the result is returned to the PLC (see Fig. 5).

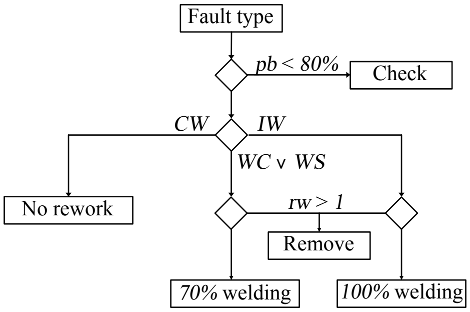


Fig. 6: Decision tree for automated rework.

H. Results

To evaluate the described system, the average times of the individual modules, the communication channels, the total time for a pin, the total time for a stator, as well as the saved time are analyzed in the following (see Fig. 7). For this purpose, series of measurements over 10 stators with 54 pins each were recorded on a standard industrial PC (Siemens SIMATIC IPC427D). The respective start and end times were stored in an external database using time stamps and evaluated afterwards. The resulting values are listed in TABLE III.

The times ΔT_{Pin}, $\Delta T_{Preprocessing}$ (ΔT_{Pre}), $\Delta T_{Prediction}$ (ΔT_{Pred}), $\Delta T_{Decision}$ (ΔT_{Dec}), $\Delta T_{preprocessingToprediction}$ (ΔT_{ptop}) and $\Delta T_{predictionTodecision}$ (ΔT_{ptod}) are the differences between the individual timestamps. The comparably small standard deviations indicate a good stability of the system, which is particularly important in production. The preprocessing module can be identified as the main consumer. With an average processing time of $0.7796s$ it represents the largest part of the chain. This result is expected, because the 3D-images are scaled down and transformed as explained in section III-C. However, it also represents the greatest potential for optimization.

A consideration of the communication times ΔT_{ptop} and ΔT_{ptod} shows that the modular distribution of the individual tasks generated only little extra time and thus strongly limits the disadvantage of the increased communication time presented in Section IV-B.

The cycle time for a complete stator ΔT_{Stator} is calculated from the difference between the last timestamp of the last pin and the first timestamp of the first pin (see Equation 1).

$$\Delta T_{Stator} = T_{P[54]_{Last}} - T_{P[1]_{First}} \tag{1}$$

A relatively small standard deviation can also be determined here, which, as with the individual modules, indicates a stable process in the overall system. It should be noted that the preprocessing processes two images at once, which results in a two-fold reduction of the throughput time compared to the sum of the times for the individual pins.

The value ΔT_{Saved}, indicates the average time saved by modularization and the resulting parallelization of the individual tasks. As described in Equation 2, the first time stamp of the i+2nd pin is subtracted from the last time stamp of

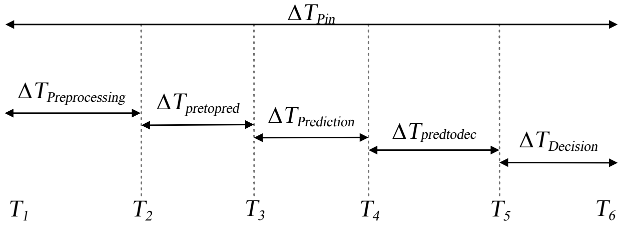


Fig. 7: Measuring points for determining the processing times.

TABLE III: STATISTICS OF THE TIME DIFFERENCE BETWEEN THE PROCESS STAGES

	ΔT_{Stator} (s)	ΔT_{Pin} (s)	ΔT_{Pre} (s)	ΔT_{Pred} (s)
Mean	21.3501	0.7968	0.7796	0.0033
Standard deviation	0.0415	0.0083	0.0070	0.0003
Min	21.3059	0.7744	0.7608	0.0026
Max	21.4231	0.8291	0.8092	0.0046

	ΔT_{dec} (s)	ΔT_{ptop} (s)	ΔT_{ptod} (s)	ΔT_{saved} (s)
Mean	0.0001	0.0086	0.0052	0.0064
Standard deviation	0.0000	0.0034	0.0013	0.0042
Min	0.0000	0.0044	0.0036	-0.0023
Max	0.0005	0.0158	0.0137	0.0189

the i-th pin. So, we compare the last timestamp of the first pin with the first timestamp of the third pin. The second pin must be skipped because it was preprocessed at the same time as the first pin by parallelization. It should be noted that the communication times were included in the calculation and therefore ΔT_{saved} represents a pure net gain.

$$\Delta T_{Saved} = T_{P[i]_{last}} - T_{P[i+2]_{first}} \tag{2}$$

A classification into the production technical context and the given application case of hairpin welding allows the following conclusions to be drawn:

V. CONCLUSION AND FUTURE WORK

This paper proposed a system that detects, classifies, and automatically reworks faults in a hairpin welding process. The hairpin welds are optically checked by a CNN that inputs 3D data and BW images. The second part of the paper presented our edge– and cloud-architecture for plant-oriented data processing. After discussing this architecture in a model-based fashion , the individual modules were presented and tested against the requirements. The modular design undercut the required cycle time ($30s$) by a factor of 0.7, thus profiting the production.

In future projects, we hope to develop an automated cloud-based system that trains and evaluates the CNN. This development would add a second optimization control loop to the current closed loop, creating a CNN that adapts to changing environmental parameters.

REFERENCES

[1] N. Sammaknejad, Y. Zhao, and B. Huang, "A review of the Expectation Maximization algorithm in data-driven process identification," *Journal of Process Control*, vol. 73, pp. 123–136, 2019.

[2] Y. Wang, Y. Si, B. Huang, and Z. Lou, "Survey on the theoretical research and engineering applications of multivariate statistics process monitoring algorithms: 2008-2017," *The Canadian Journal of Chemical Engineering*, vol. 96, pp. 2073–2085, 2018.

[3] A. Kampker, *Elektromobilproduktion (german), Electric vehicle production (english)* . Berlin: Springer Vieweg, 2014.

[4] J. Vater, P. Schamberger, A. Knoll, and D. Winkle, "Evaluation of Machine Learning for Quality Monitoring of Laser Welding Using the Example of the Contacting of Hairpin Windings," in *2019 9th International Electric Drives Production Conference (EDPC)*. IEEE, 2019.

[5] J. Um, V. Gezer, A. Wagner, and M. Ruskowski, "Edge Computing in Smart Production." Springer International Publishing, 2020, vol. 980, pp. 144–152.

[6] D. Mery, "Aluminum Casting Inspection Using Deep Learning: A Method Based on Convolutional Neural Networks," *Journal of Nondestructive Evaluation*, vol. 39, p. 14, 2020.

[7] Z. Mo, L. Chen, and W. You, "Identification and Detection of Automotive Door Panel Solder Joints based on YOLO," in *2019 Chinese Control And Decision Conference (CCDC)*, 2019, pp. 5956–5960.

[8] C. G. Pachón S., J. Pinzón Arenas, and R. Jiménez Moreno, "Detection of Scratches on Cars by Means of CNN and R-CNN," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 9, p. 745, 2019.

[9] M. Weigelt, A. Mayr, J. Seefried, P. Heisler, and J. Franke, "Conceptual design of an intelligent ultrasonic crimping process using machine learning algorithms," *Procedia Manufacturing*, vol. 17, pp. 78–85, 2018.

[10] A. Mayr, B. Lutz, M. Weigelt, T. Gläßel, D. Kißkalt, M. Masuch, A. Riedel, and J. Franke, "Evaluation of Machine Learning for Quality Monitoring of Laser Welding Using the Example of the Contacting of Hairpin Windings," in *2018 8th International Electric Drives Production Conference (EDPC)*. IEEE, 2018, pp. 1–7.

[11] J. Vater, P. Schlaak, and A. Knoll, "A Modular Edge-/Cloud-Architecture in automotive production for Automated Error Detection and Correction using Machine Learning," in *IEEE Computer Society Signature Conference on Computers, Software and Applications (COMPSAC)*, 2020, in press.

[12] A. Riedl, M. Manusch, M. Weigelt, T. Gläßel, A. Kühl, S. Reinstein, and J. Franke, "Challenges of the Hairpin Technology for Production Techniques," in *2018 21st International Conference on Electrical Machines and Systems (ICEMS)*, 2018, pp. 2471–2476.

[13] A. Blom, P. Dunias, P. van Engen, W. Hoving, and J. de Kramer, "Process spread reduction of laser microspot welding of thin copper parts using real-time control," in *Photon Processing in Microelectronics and Photonics II*, 2003.

[14] A. Heider, P. Stritt, A. Hess, R. Weber, and T. Graf, "Process Stabilization at welding Copper by Laser Power Modulation," *Physics Procedia*, vol. 12, 2011.

[15] V. D. I. (VDI), *Entwicklungsmethodik für mechatronische Systeme (VDI 2206): Design methodology for mechatronic systems.* VDI, 2004.

[16] *Cyber-Physical Systems: Innovation durch softwareintensive eingebettete Systeme*. Springer, 2010.