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 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

consistent wording

Table 1: Industry 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: Industry 4.0 demostration cell: Controlled Use-Cases. Use-Cases with the value *Robot position: NOK* and *Conveyor speed: Too Fast* will result in NOK pieces.

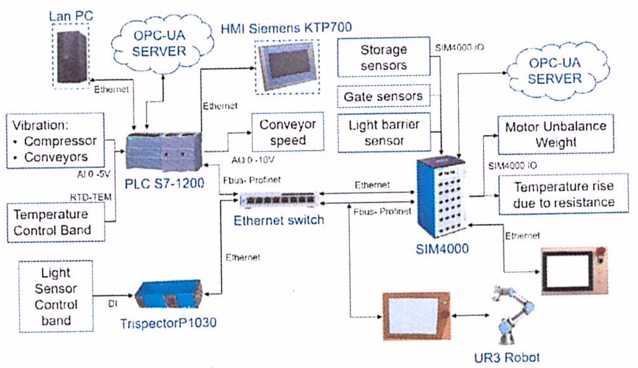

Fig. 2: Industry 4.0 demostration cell: Connection diagram (simplified).

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 [3]. The raw data collected from the sensors is usually tagged by an alphanumeric code predefined by the sensor manufacturer. 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 human-readable words 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 was clean and compiled in one directory, the next 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 tag* included in the dataset Table 1. Subsequently, each data-sample was automatically checked for missing sensor values and discarded if so.

Each time-series sample was the result of data collected during one machine assembly cycle (MAC). The starting point of one MAC was defined as the very first moment the robotic arm grabs the white disk from the store area and the ending point was defined as the starting of the subsequent MAC, the grabbing of the next white disk in the same area. To eliminate time as a variable in our dataset, it was needed to transform time-series into feature data suitable describing a MAC in a way that different MACs are easily comparable. For that reason, we defined a set of sampling-conditions that helped us to collect the relevant data during the MAC:

- Value of the robot position (xyz) while dropping both disks in the conveyor - Gate
- 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 this sampling-condition, 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 was framed as a classification task. The input vectors which we fed 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 was used as the training target. It can be OK or NOK which we encoded as zero and one.

We worked with a total of 528 measurements. Each containing arm and belt data logs of individual cell run. 50 samples were set aside at random for testing purposes, leaving 478 training samples. The random number generator seed was set to one piece, and this might compensate for disadvantages of the statistical methods.

, see p. 9.

7. Lesson Learned

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. Difficulties at implementing ML models in production are also described by others authors [15]. **Incompatibility between technologies**; devices configured on the same network presented troubles to share data 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 led to a series of data synchronization problems due to differences in the internal clock of both servers. 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 complicated, time-consuming, and a fundamental step for 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 occur, made it complicated to have accurate data sampling from each assembly cycle. To overcome this uncertainty, we modified the machine program to raise specific flags whenever certain actions were executed.

Machine Learning; careful measurement and meaningful preprocessing of our cell-data 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 [7]. 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.

8. Author Contributions

Pérez M. and Moritz performed writing - original draft, conceptualization, review, and editing; Investigation; Software and Data Curation. Schmallenbach performed writing - Review & editing. Wagner, Heinbach, Steinberg, Steffes-lai, Garcke performed writing - Review. Burggräf performed supervision of work.

Acknowledgments

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

References

[1] Aggarwal, C.C., 2014. Data classification. CRC PRESS, 2020.

[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] Burggräf, P., Wagner, J., Koke, B., 2018. Artificial intelligence in production management: A review of the current state of affairs and research trends in academia .

[5] Delen, D., Demirkan, H., 2013. Data, information and analytics as services. Decision Support Systems 55.

[6] Escobar, C.A., Morales-Menendez, R., 2018. Machine learning techniques for quality control in high conformance manufacturing environment. Advances in Mechanical Engineering 10.

[7] Fisher, R.A., 1936. The use of multiple measurements in taxonomic problems. Annals of Eugenics 7, 179–188.

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

[9] Hinckley, C.M., 1997. Defining the best quality-control systems by design and inspection. Clinical Chemistry 43.

[10] 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.

[11] 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.

[12] Krauß, J., Pacheco, B.M., Zang, H.M., Schmitt, R., 2020. Automated machine learning for predictive quality in production. Procedia CIRP 93.

[13] Marsland, S., 2015. Machine Learning: An Algorithmic Perspective. 2 ed., CRC Press.

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

[15] Paleyes, A., Urma, R.G., Lawrence, N.D., 2020. Challenges in deploying machine learning: a survey of case studies .

[16] Ritter, C., Schweitzer, F., 1992. Neue wege der prozeßmodellierung für die rechnergestützte qualitätsvorhersage und -sicherung in der spanplattenfertigung. Holz als Roh- und Werkstoff 50.

[17] Robinson, H., 1997. Using poka-yoke techniques for early defect detection .

[18] 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.

[19] Selvamuthu, D., Das, D., 2018. Introduction to Statistical Mehods, Design of Experiments and Statistical Quality Control. Springer.

[20] Sohnius, F., Schlegel, P., Ellerich, M., Schmitt, R.H., 2019. Data-driven prediction of surface quality in fused deposition modeling using machine learning .

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