Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 138 (2026) 1073–1078

www.elsevier.com/locate/procedia

18th CIRP Conference on Intelligent Computation in Manufacturing Engineering

Generalized Statistical Process Control via 1D-ResNet Pretraining

Tobias Schulze[a,*], Louis Huebser[a], Sebastian Beckschulte[a], Robert H. Schmitt[a,b]

[a] *Laboratory for Machine Tools and Production Engineering, WZL of RWTH Aachen University, Campus-Boulevard 30, 52074 Aachen, Germany*

[b] *Fraunhofer Institute for Production Technology IPT, Steinbachstraße 17, 52074 Aachen, Germany*

* Corresponding author. Tel.: +49-241-80-22293. *E-mail address:* tobias.schulze@wzl-iqs.rwth-aachen.de

Abstract

Statistical Process Control (SPC) suffers from high false positive rates for non-normally distributed quality characteristics as stability criteria are too sensitive. This makes SPC uneconomic when applied consequently due to production downtimes. To overcome limitations of SPC, we develop an approach limiting the false postitives without changing the quality inspection workflow. Based on synthetic data subject to an approach-specific definition of stability, a 1D-Residual Neural Network (1D-ResNet) is pretrained. The pretrained model can subsequently be applied to various use cases without the need of large amounts of data. A benchmark against SPC shows a significant decrease in false positives.

Peer-review under responsibility of the scientific committee of the 18th CIRP Conference on Intelligent Computation in Manufacturing Engineering (CIRP ICME '24)

Keywords: Statistical Process Control; Predictive Quality; Process Monitoring; Process Stability; Anomaly Detection

1. Introduction

Processes deteriorate over time due to random and systematic deviations, leading to increased scrap rates. It is, therefore, necessary to monitor processes and take corrective action in case of deviations. Statistical process control (*SPC*) is a widespread method for process monitoring as a more efficient alternative to 100% inspection. However, the approach tends to raise false alarms and relies on a normal distribution of the analyzed quality characteristics. [1-3]

Predictive quality (*PQ*) can address the shortcomings above by predicting quality characteristics using machine learning (*ML*) models based on machine and sensor data [4]. This way, inspection frequencies can be reduced while maintaining a virtual 100% inspection, replacing *SPC* [5]. However, this approach does not enable the prediction of process performance and, consequently, the quality of future-produced parts.

Based on the outlined situation, the main objective of this paper is to propose a new approach for process control in discrete manufacturing processes that overcomes the limitations of *SPC* in terms of false alarms based on monitoring the part quality predictions to predict process deteriorations[1].

After a short introduction to process control, section two outlines the concept behind the proposed approach and an approach-specific redefinition of process stability is introduced in section three. The approach is detailed in section four and evaluated in section five by a benchmark against *SPC*. The results are discussed, and further work is defined in section six.

[1] Source Code available at: https://doi.org/10.5281/zenodo.10973131
Synthetic datasets available at: https://doi.org/10.5281/zenodo.8249487

2212-8271
Peer-review under responsibility of the scientific committee of the 18th CIRP Conference on Intelligent Computation in Manufacturing Engineering (CIRP ICME '24)
10.1016/j.procir.2026.01.185

2. Technical foundations

Early detection of process deviations requires continuous process monitoring [6,7]. *SPC* relies on detecting process instabilities by analyzing measurements of quality characteristics for randomly sampled products within the production line [1,8].

A stable process is a requirement for the usage of *SPC*, which means that the process is only affected by random-sourced variation. Stability criteria are defined to determine the stability of the process. The most basic stability criteria considered in *SPC* are: exceeding the control limits, a trend (more than seven consecutive values in ascending or descending order), and a run (more than seven consecutive values on one side of the mean value). The control limits are calculated based on the historical variation and location of the process and depend, among other factors, on the desired quality level (it is usually required that 99.73% of the measured values are within the control limits). If the process is stable, the next step is determining its capability. For this purpose, the position and dispersion of the process results are compared to the tolerance specification. The actual quality level of the process is determined based on the determined capability parameters. If this level meets the requirements, the process can be monitored using quality control charts. In that context, it is monitored whether the stability criteria are violated. After each violation, the control limits are recalculated. Besides, process capability is continuously evaluated. [1,8-11]

In addition to the measured quality characteristics used in univariate *SPC*, multivariate *SPC* takes recorded process characteristics into account to monitor the process [12]. The prerequisite for this is that the values of the individual process parameters are normally distributed, which is rarely the case in practice [13]. Moreover, the performance decreases as the number of analyzed process parameters increases [8]. Another limitation is that only patterns defined in advance, such as the stability criteria of *SPC*, can be recognized. This also holds for most *ML* research in this context, as it uses the models to identify predefined patterns. [14-16]

Based on the described limitations, *PQ* can support predicting quality characteristics for products that are not physically measured using *ML* models trained on process data [4,5,9,17]. Examples of *PQ* applications include defect detection in an automotive application or determining the influence of various process parameters on the quality of additively manufactured components [18]. The knowledge gained enables statements about the quality of the manufactured products already during production or to reduce physical testing up to complete virtual testing [5,19].

ML models commonly used for *PQ* are suitable for recognizing non-explicitly defined patterns and do not rely on the data of the individual process parameters being normally distributed [20]. Nevertheless, these mere quality predictions only reflect the current process status but do not allow process changes to be detected in advance [19]. For a specific use case in the automotive industry, BECKSCHULTE ET AL. developed an approach based on detecting changes in the distributions of the quality characteristics values before faults occur [2].

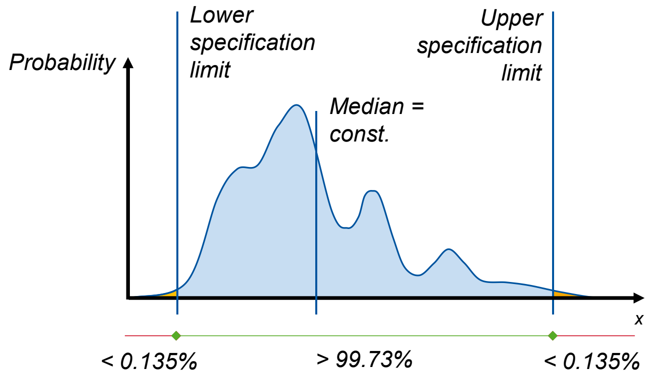

Fig. 1. Approach-specific redefinition of process stability.

3. Process monitoring via distribution monitoring

Our approach builds on the BECKSCHULTE ET AL.'s idea of detecting deteriorations via the probability distributions of the quality characteristics. Compared to the mentioned approach, the developed approach is not specific to one use case, but the detection model is trained on synthetical data. The trained model can then be used independently of the application. The general concept is outlined in section 3.1. The underlying definition of stability is given in section 3.2.

3.1. Concept for distribution-based process monitoring

In industrial production, changes in process performance do not usually occur suddenly, but the process deteriorates over time. For instance, wear impacts the perfomance of the process even before the tool breaks. The basic idea of our approach is that such gradual changes affect the probability distribution of the quality characteristic values. Our approach aims to recognize these changes in the probability distribution of the process outputs independently of the actual probability distribution and the quantity of process parameters. Accordingly, we call our approach Generalized Statistical Process Control (*G-SPC*). It relies on the idea that a finite number of probability distributions refers to a process under control, called *OK* distributions. Conversely, there are infinite distributions corresponding to a process that is out of specifications, called *NOK* distributions. A new definition of stability is introduced in the following section to define the requirements for the *OK* distributions.

3.2. Approach-specific redefinition of process stability

A probability distribution must correspond to a stable process for it to be considered an *OK* distribution. Process stability is redefined to meet the requirements of the proposed approach. A process is considered stable if, on the one hand, the median is constant and, on the other hand, a defined percentage of the values mapped by the quantiles of the probability distribution lies within the tolerance limits specified for the process as shown in Fig. 1. This eliminates the control limits specified by the *SPC*, which already require intervention without the tolerance limits being violated. The median is chosen as the location parameter, as the mean value is not defined for several types of probability distributions. The choice of quantile values depends on the desired process capability level. It is selected for this approach so that 99.73% of the values lie within the tolerance interval (as for the control limits given by *SPC*).

4. The *G-SPC* approach

The approach comprises three steps. In the first step, synthetic data is generated to pretrain the *G-SPC* prediction model. Section 4.1 explains how this dataset is generated to represent gradually changing production processes over time accurately. Section 4.2 discusses the selection of a suitable architecture for the *G-SPC* prediction model and its training process. After training, the model can predict processes out of control based on actual process data (inference). This procedure is described in section 4.3. Fig. 2 shows the *G-SPC* approach.

4.1. Data Generation

Synthetic data is generated to represent the process characteristics and their corresponding quality features to mimic the behavior of an industrial process. To represent continuous process changes, a population with characteristics that vary over time is required to reflect the current situation of the process. Furthermore, the outputs of many production processes cannot be represented by a single distribution type; instead, they are a superposition of different probability distributions. Accordingly, the generation of the synthetic data is based on a mixture distribution, which is composed of several probability distributions. The weighting of the individual distributions in the mixture distribution is variable over time to depict the process changes.

To model the *OK* phase, in the first step, initial weights of individual probability distributions within the mixture distribution at the various points in time are set, as depicted in Fig. 3. To map the gradual change of the process, linear interpolation is used between the initialized weights. The remaining weights are determined by solving an optimization problem for each time step. It aims to minimize the pairwise squared differences between the weights of all the probability distributions and thus achieve a mixture distribution that is as diverse as possible. The weights sum up to 1. As a constraint, the mixture distribution must fulfill the stability definition from section 3.2, while the individual probability distributions may vary in the median. Each product (one row in Fig. 3) represents a snapshot of the process and, therefore, its probability distribution at a discrete point in time. Accordingly, a random sample of size one is drawn for each time step, representing the product's quality characteristic (x). Afterward, these values of the different products are concatenated, forming a time series.

To simulate a process that gradually transitions to a *NOK* state, the previously generated mixture distribution is superimposed with a distribution that does not fulfill the stability criteria defined in this paper with linearly increasing weighting. The distribution is chosen so that the resulting distribution during the ramp-up also transitions to a state that does not satisfy the stability definition.

According to the assumptions on which this paper is based, production processes are initially stable after set-up before they gradually leave this state due to degradation. If such a deviation is detected, the process is adjusted until it is back in a stable state and gradually degrades again. In order to map this cyclical behavior, the synthetic data is also composed cyclically from the *OK* and *NOK* phases. The *OK* phases are chosen longer than the *NOK* phases as processes usually operate longer in a stable state than it takes to get them back into this state again after they degrade.

Next to the value sampled at each time step (x), whether this value is within the limits (y) is also stored. In addition, the distribution weights implicitly record the phase (*OK*/*NOK*) in which the value was generated. For this paper, the training dataset was generated with 100,000 cycles of 100 data points with 27 randomly parametrized *OK* normal distributions, 9 of them centered. For each cycle a *NOK* normal distribution was parametrized randomly. Around each cycle change, there is a *NOK* phase of random length, leading to a total of 10.74% values in the *NOK* phase.

4.2. Pretraining

The input x for the *G-SPC* model and the output y, which is used to train the anomaly score, are time series data. Due to the type of modeling, they implicitly contain information on the mixture distribution at the respective point in time.

Just as there are strong correlations between neighboring pixels in images, there are also strong correlations between successive values from time series. Accordingly, image

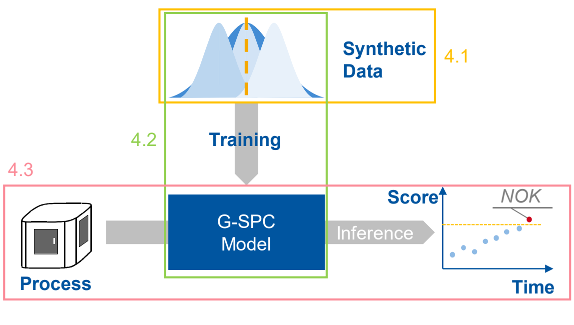

Fig. 2. The *G-SPC* approach.

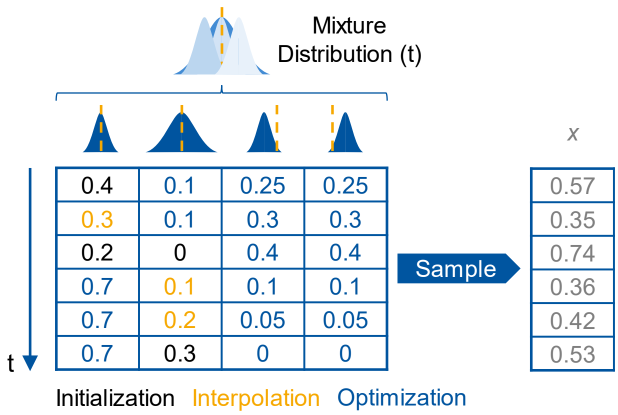

Fig. 3. Determination of the weighting of different probability distributions within the mixture distribution during *OK* phase and random sampling of one value x at each time step t.

processing models are also suitable for analyzing time series, although there are only correlations in one direction (time) instead of two (x and y axes). [10,21] Convolutional neural networks (*CNN*), in particular, are used for this purpose in the literature as it is more straightforward to formulate the problem for such encoder approaches than for sequence approaches like long short-term memory networks [2,22-23]. Even better results than basic *CNNs* are promised by residual neural networks (*ResNet*), which can further reduce error rates due to their internal structure [24]. For example, MA ET AL. use a *CNN* to detect anomalies in continuous production by monitoring the time series of process parameters [25].

Therefore, several *1D-CNN* and *ResNet* structures were tested as models for *G-SPC*. As most processes have a relatively low error rate compared to the samples produced, there is an imbalance between *OK* and *NOK* data. Hence, the squared differences are weighted by taking the actual output plus one to the power of four to calculate the mean square error as the loss function of the model.

The model is to be trained with regard to the probability of occurrence of an error, not a pure classification. The training data must be preprocessed for this purpose. Values between 0 (lowest probability of error) and 1 (highest probability of error) are selected to define the probability of occurrence. A linear ramp-up of length n precedes each faulty value to represent that limit violations usually do not happen suddenly (see Fig. 4). Both selected model structures process time series with the sliding window approach. Therefore, the data must be sliced into windows with length k. Sliding windows starting with an output bigger than 0 as their first time step are discarded for model training, as the target is to detect anomalies before they happen and not anomalies that already happened.

Model training showed that the number of layers and their structure had no significant impact on the model training error. At the same time, the results of *ResNet* significantly outperformed the *CNN* on the test dataset (c.f. section 5.1). As a result, a *ResNet* was selected as the final *G-SPC* model. The best results were achieved with one identity and three convolution blocks, each followed by two more identity blocks. These blocks are followed by an average pooling and two fully connected layers with a linear activation function resulting in one scalar value. It turned out that the best results were obtained using a ramp length of $n = 3$ and a window length of $k = 18$ for data preprocessing. Also, scaling the input data and target to the interval [0, 0.5] and subtracting the mean from the input data improved the results.

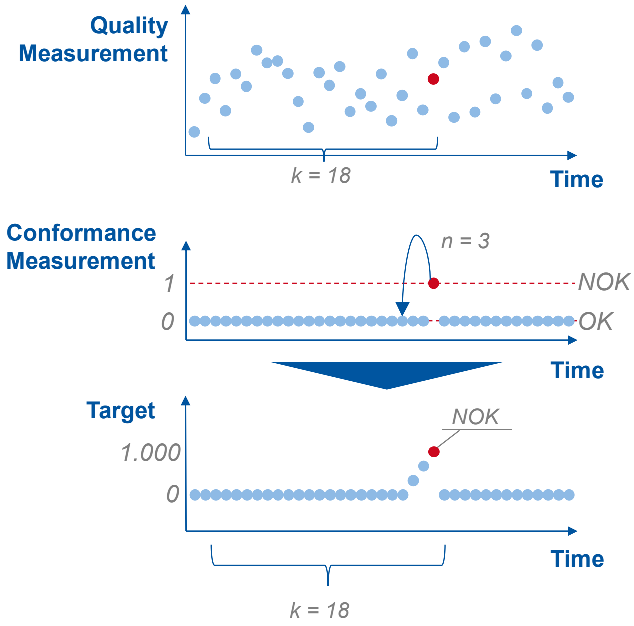

Fig. 4. Data preprocessing steps for *G-SPC*.

4.3. Model application

After the pretraining, the model can be deployed in industrial applications. For this purpose, the data must be preprocessed the same way as the training dataset. Based on the last sliding window, the model provides the anomaly score associated with the current product. If the anomaly score exceeds a use-case-specific predefined threshold, the process is considered out of control, and corrective action should be taken before continuing production.

5. Evaluation

The evaluation of *G-SPC* is done using a two-step approach. First, the model is benchmarked against a traditional *SPC* on a synthetically generated test dataset. In the second step, the applicability is evaluated in an actual use case from the industry.

5.1. Benchmark analysis

For testing the model, a test dataset was generated as defined in section 4.1, this time comprising more than one type of *OK* distributions (normal, Laplace, and uniform). Again, the median was set to 0.5 with a lower control limit of 0 and an upper control limit of 1. This time, 20 cycles with length 1000 each were generated. For the *G-SPC*, data were preprocessed in the same way as described in section 4.2. The anomaly threshold was set to 0.1, as it led to the best results. The *Shewart Individual Chart* was selected for SPC, as sampling is unnecessary for a *PQ* setting with predictions for all instances. The control limits for *SPC* were calculated based on the values in the first *OK* phase of the test dataset.

For assessing the *SPC*, violations of the stability criteria (violation of limits, trend, run) were considered as predicted positive, while violations of the tolerance limits were considered as actual positive. As *G-SPC* uses sliding windows and on synthetic datasets, information is available on whether the value stems from an *OK* or *NOK* phase, the definition needs to be slightly adjusted, as given in Table 1. For both *SPC* and *G-SPC*, true negatives are not defined as given by the theoretical considerations after the first alarm, all subsequent time steps up to the occurrence of the actual fault should also emit signals which is not necessarily given [2].

Table 1. Definition of the Confusion Matrix fields for *G-SPC* with A: predicted signal from *G-SPC* (1 if anomaly score is below threshold), B: phase information (1 if all datapoints within sliding window stem from *OK* phase), C: violation of tolerance limits (1 if no violation of tolerance limits within sliding window).

	A	B	C
True positive	0	0	0
False positive	0	1	0 or 1
	0	0	1
False negative	1	0	0
True negative	Not defined	Not defined	Not defined

Performance is measured by recall to evaluate whether the models detect all violations of tolerance limits (within the *NOK* phase) and precision to evaluate whether the predictions as anomalies are actually violations of the tolerance limits (within the *NOK* phase). The F1-score is then used to evaluate whether both targets are met at the same time.

Table 2. Evaluation metrics of synthetic test dataset.

Metric	*SPC*	*G-SPC (CNN)*	*G-SPC (ResNet)*
Recall	**1.00**	0.359	0.743
Precision	0.571	**0.896**	0.856
F1-score	0.727	0.513	**0.795**

The results listed in Table 2 show that *SPC* performs best in terms of recall. In contrast, *G-SPC* using the *CNN* model performs best in precision, and *G-SPC* using *ResNet* has the highest F1-score. The F1-score of *G-SPC* with *CNN* is significantly lower than that of *SPC*, and its precision is only 4% lower than that of *G-SPC* with *ResNet*. For this reason, as mentioned in section 4.2, *CNN* is not considered in the remainder of this paper.

On the one hand, while *SPC* detects all limit violations, 42.9% of its alarms are false, indicating no actual limit violation. On the other hand, *G-SPC* (from now on with *ResNet*) misses 25.7% of the situations where it should detect process deterioration. However, it has a much lower rate of false alarms of 14.4%, leading to fewer unnecessary process interruptions. Since *PQ* performs a 100% virtual quality measurement, it is not critical that not all deteriorations are detected, as this does not result in a *NOK* product passing the quality inspection.

Considering the confusion matrices in Fig. 5, the total number of false alarms (false positives) using *G-SPC* is less than half as high (1,015 vs. 2,173) as when using the traditional *SPC* on the same dataset. Nevertheless, the total number of actual positives for the *G-SPC* approach is almost three times higher (8,092 vs. 2,887) than for the traditional *SPC*. This is because any sliding window containing a limit violation in the *NOK* phase is considered an actual positive when using *G-SPC*. This implies that when using the model, the production process must be readjusted before process monitoring is restarted.

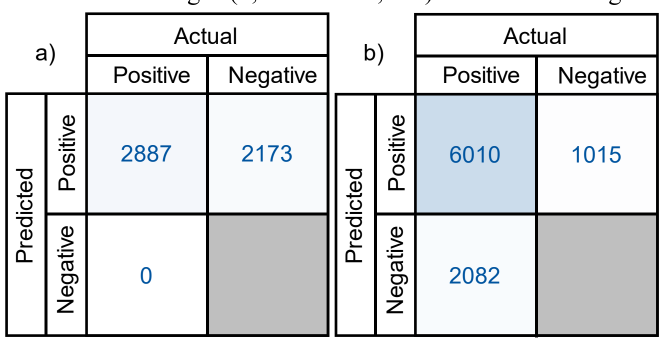

Fig. 5. Confusion matrix of synthetic test dataset. a) *SPC*; b) *G-SPC* (*ResNet*)

5.2. Application to electric control screw driving process

In the industry use case, an automotive OEM's safety-relevant manufacturing process step is selected, consistent across all vehicles despite their high variability. The process involves attaching the seatbelt anchor to the vehicle frame on the left-hand side. The attachment is performed using an electric control screwdriver that records the torque during the fastening process. These recorded torque values are compared with the vehicle-specific target values. If the torque deviates from the specified limits, the fastening process is repeated to ensure the vehicle meets safety standards before moving to the next assembly step. The process is deemed compliant, and the vehicle is cleared for the next stage only if the torque is within the acceptable range; otherwise, the fastening is redone. The dataset reveals that 4.4% of all measurements fall outside the torque compliance range, indicating the critical nature of this process step for quality control.

As the dataset contains extreme outliers, the control limits for *SPC* were calculated based only on the values within tolerance limits. The data preparation was performed for *G-SPC* as described in section 4.2, except for the input scaling. The lower tolerance limit value was set to 0, and the upper limit to 1. This results in a not-centered median of 0.628.

Table 3. Evaluation metrics of use-case.

Metric	*SPC*	*G-SPC*
Recall	**1.00**	0.587
Precision	0.317	**1.00**
F1-score	0.481	**0.740**

The assessment of *G-SPC* was altered compared to section 5.1, as the phase information regarding *OK*/*NOK* is not available for real industry data. Anomaly scores above the threshold were considered predicted positive while sliding windows with violations of the tolerance limits inside were considered actual positive.

As already observed in the benchmark on synthetic data, *G-SPC* again outperforms *SPC* in terms of precision and

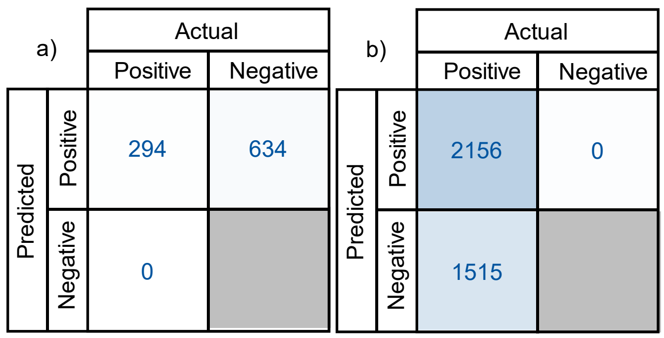

Fig. 6. Confusion matrix of the use case dataset. a) *SPC*; b) *G-SPC*

F1-score, while *SPC* has a better performance regarding recall (c.f. Table 3). *SPC* detects all actual limit violations, as shown in Fig. 6, but the precision of 0.317 is relatively low (c.f. Table 3). *G-SPC* detected only 58.7% of the sliding windows with limit violations. Again, this is not critical for its application. While *G-SPC* has no false alarms, 68.7% of *SPC* alarms are false. *G-SPC* should trigger in 55.7% of all 7,092 time steps, while this is only the case in 4.4% with *SPC* due to the different definitions of actual positives. The significant difference corresponds to the limit violations being almost equally distributed over the whole period under consideration. This shows that *G-SPC* should only be applied to initially stable processes with low error rates.

6. Conclusion, discussion, and further work

The benchmark against synthetic and real data shows that *G-SPC* significantly reduces false positives compared to traditional *SPC*, although at the cost of increasing false negatives. However, this is not critical when applied to virtual quality measurements, as scrap detection is performed beforehand. Hence, *G-SPC* can be a valuable tool for process monitoring. It extends *PQ*-based virtual quality measurements and outperforms traditional *SPC*, especially when measures are taken to stabilize the process and restore its capability after detecting process deteriorations.

The evaluation results indicate that the model does not need to be retrained for different probability distributions and different medians. Nevertheless, further investigations are required here, particularly as the phase information is not available for the real dataset with the different median. Furthermore, it should be noted that different conditions for actual positives and negatives had to be defined for *SPC* and *G-SPC* as part of the evaluation due to their characteristics. Although *G-SPC* performs better than *SPC* in the evaluation, there is no methodology to prove that the model actually makes predictions based on the probability distribution.

Despite the achieved results, further parameter constellations should be tested to improve the model's performance. For example, instead of the linear ramp-up in data preprocessing, the actual weights of the distributions within the mixture distribution could be considered, or the length of the ramp-up could be varied. Furthermore, different loss functions, window lengths, and network architectures should be tested.

In addition to the potential improvements mentioned, the model should be tested in more real-world use cases. Particularly, when a deterioration is detected, the process should be adjusted before it is monitored again. Regarding the synthetic dataset, additional distribution types and variable cycle lengths could be tested, and data generation could be used to build realistic big-data benchmark datasets for *SPC*.

Acknowledgements

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy – EXC-2023 Internet of Production – 390621612.

References

[1] Schmitt RH, Pfeifer T. Qualitätsmanagement. Strategien – Methoden – Techniken. 5th ed. München: Hanser; 2015.

[2] Beckschulte S, Huebser L, Klasen N, et al. Potenziale von Neuronalen Netzen gegenüber SPC zur Fehlervermeidung in der Prozesssteuerung. In: Woll R., Goldmann C, editors. Trends und Entwicklungstendenzen im Qualitätsmanagement. GQWJT 2021. Wiesbaden: Springer.

[3] Hryniewicz O. SPC of Processes with Predicted Data: Applications of the Data Mining Methodology. In: Knoth S, Schmid W, editors. Frontiers in Statistical Process Control 11. Cham: Springer; 2015.

[4] Schmitt RH, Kurzhals R, Kiesel R, et al. Predictive Quality – Data Analytics zur Steigerung unternehmerischer Nachhaltigkeit. In: Bergs T, et al, editors. Internet of Production - Turning Data into Sustainability: AWK'21. 1st ed. Aachen: Apprimus;2021:289-318.

[5] Cramer S, Huber M, Knott A, Schmitt RH. Wertschöpfung in Industrie 4.0: Virtuelle 100 %-Prüfung durch Predictive Quality. Zeitschrift für wirtschaftlichen Fabrikbetrieb 2023; 118(5):344-9.

[6] Ismail M, Mostafa NA, El-assal A. Quality monitoring in multistage manufacturing systems by using machine learning techniques. J Intell Manuf 2022; 33:2471–86.

[7] Yin S, Ding X, Xie X, Luo H. A Review on Basic Data-Driven Approaches for Industrial Process Monitoring. IEEE Transactions on Industrial Electronics 2014; 61(11):6418-28.

[8] Woodal W, Montgomery C. Research Issues and Ideas in Statistical Process Control. Journal of Quality Technology 1999; 31(4):376-86.

[9] Schmitt RH, Pfeifer T. Masing Handbuch Qualitätsmanagement. 7th ed. München: Hanser; 2021.

[10] DIN ISO 22514-1. Statistical methods in process management – Capability and performance –1: General principles and concepts; 2014.

[11] ISO 7810-1. Control charts – Part 1: General guidelines; 2019.

[12] Bersimis S, Panaretos J, Psarakis, S. Multivariate Statistical Process Control Charts and the Problem of Interpretation: A Short Overview and Some Applications in Industry. Proceedings of the 7th Hellenic European Conference on Computer Mathematics and its Applications; 2005.

[13] Mason RL, Champ CW, Tracy ND, et al. Assessment of multivariate process control techniques. J Qual Technol 1999; 29(2):140-3.

[14] Zorriassatine F, Tannock J. A review of neural networks for statistical process control. Journal of Intelligent Manufacturing, 1998; 9:209-24.

[15] Zan T, Liu Z, Wang H, et al. Control chart pattern recognition using the convolutional neural network. J Intell Manuf 2020; 31:703-16.

[16] Zan T, Liu Z, Su Z, et al. Statistical Process Control with Intelligence Based on the Deep Learning Model. Applied Sciences 2020; 10(1):308.

[17] Beckschulte S, Günther R, Huebser L, Schmitt RH. Mit Predictive Quality in die Zukunft sehen. Zeitschrift für wirtschaftlichen Fabrikbetrieb 2020;115(10):715–8.

[18] Buschmann D, Enslin C, Elser H, et al. Data-driven decision support for process quality improvements. Procedia CIRP;99:313-8.

[19] Buschmann D, Schulze T, Enslin C, Schmitt RH. Interpretation Framework of Predictive Quality Models for Process- and Product-oriented Decision Support. Procedia CIRP;118:1066-71.

[20] Beckschulte S, Klasen N, Huebser L, Schmitt RH. Prädiktive Qualität in der Prozesslenkung: Neuronales Netz als SPC 4.0. Zeitschrift für wirtschaftlichen Fabrikbetrieb 2021; 116(10): 662-6.

[21] Lecun Y, Bengio Y. Convolutional networks for images, speech, and time-series. In: Arbib MA, editor. The handbook of brain theory and neural networks. MIT Press; 1995.

[22] Zhao B, Lu H, Chen S, et al. Convolutional neural networks for time series classification. J. Syst. Eng. Electron. 2017; 28(1):162-9.

[23] Iwana BK, Uchida S. An empirical survey of data augmentation for time series classification with neural networks. Plos one 2021; 16(7): e0254841.

[24] He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. IEEE CVPR 2016; 770-8.

[25] Ma F, Ji C, Wang J, Sun W. Early identification of process deviation based on convolutional neural network. Chinese Journal of Chemical Engineering 2023; 56:104-18.