Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 120 (2023) 1185–1190

www.elsevier.com/locate/procedia

56th CIRP Conference on Manufacturing Systems, CIRP CMS '23, South Africa

Unsupervised anomaly detection in unbalanced time series data from screw driving processes using k-means clustering

Nikolai West[a,*], Thomas Schlegl[b], Jochen Deuse[a,c]

[a]*Institute of Production Systems, Technical University Dortmund University, Leonhard-Euler-Str. 5, 44227 Dortmund, Germany*
[b]*BMW Group, Petuelring 130, 80788 Munich, Germany*
[c]*Centre for Advanced Manufacturing, University of Technology Sydney, 15 Broadway, Ultimo NSW 2007, Australia*

* Corresponding author. Tel.: +49-231-755-2651; fax: +49-231-755-2649. *E-mail address:* nikolai.west@tu-dortmund.de

Abstract

Since bolted joints are ubiquitous in manufacturing, their effective and reliable quality assurance is particularly important. Most tightening processes rely on statistical methods to detect faulty screw connections already during assembly. In this paper, we address the detection of faulty tightening processes using a clustering based approach from the field of Unsupervised Machine Learning. In particular, we deploy the *k*-Means algorithm on a real-world dataset from the automotive industry. The model uses Dynamic Time Warping to determine the similarity between the normal and abnormal tightening processes, treating each one as an independent temporal sequence. This approach offers three advantages compared to existing supervised methods: 1.) time series with different lengths can be utilized without extensive preprocessing steps, 2.) errors never seen before can be detected using the unsupervised approach, and 3.) extensive manual efforts to generate labels are no longer necessary. To evaluate the approach, it is applied in a scenario where actual class labels are available. This allows evaluating the clustering results using traditional classification scores. The approach manages to achieve an accuracy of up to 88.89% and a macro-average F1-score of up to 63.65%.

Keywords: Anomaly detection; time series clustering; unsupervised learning; screw driving; machine learning; tightening data; manufacturing; open source

1. Introduction

The digitization of production systems is steadily increasing the amount of information and communication technology used. This provides manufacturing companies with cost-effective access to advanced sensor technology and wireless communication technologies [1, 2]. The interconnectivity of assets in a digital factory accelerates the adoption of data acquisition and storage solutions. This often leads to an extensive archive of manufacturing data. Any future use of this *data* first requires the retrieval of *information* before it can become *knowledge*. Machine Learning (ML) enables such a transition and allows gaining novel insights for production planning and control [3].

Quality Management constitutes a very promising domain to leverage said data availability in manufacturing companies. Here, any deterioration in process quality may lead to increased defect rates, higher rework or scrap costs as well as reputational harm [4]. To prevent this, many manufacturing companies rely on data-driven methods for process monitoring. For almost a century, *Statistical Process Control* (SPC) has been one of the most important tools [5]. SPC control charts are frequently used to record and manage the deviation of targeted metrics in various processes [3]. However, these systems are only suitable to an extent for detecting complex or previously unseen defects [6]. In this paper, we present a novel approach to detect process anomalies in screw driving data using time series clustering.

10.1016/j.procir.2023.09.146

2. Fundamentals

2.1. Anomaly detection in time series data from screw driving

As introduced, declining costs for data collection and storage cause a significant increase in volume and velocity of industrially collected data. A common way to record data in manufacturing is in the form of *time series*. A time series is a sequence of values indexed in chronological order using time stamps. These values consist of successive measurements made from the same source over a fixed time interval and are used to track change over time [7]. As such, we also consider data from tightening processes as time series. Digital systems often record the torque and angle of rotation to monitor the physical forces during tightening. Since the angular velocity in tightening processes is typically constant, we treat the recorded data analogously to any time series.

Bolted joints are widely used in assembly processes. A typical car, for example, contains around 3,000 threaded fasteners [8]. State-of-the-art manufacturing companies analyze the data from tightening processes to ensure the desired level of product quality [9]. Traditional approaches, such as SPC, are well able to reduce process variations and to stabilize product quality. However, they are ill suited to detect abnormal processes in daily operations, where data volumes can reach thousands or millions of time series. In this context, we refer to any tightening process that deviates from the normal and expected pattern as an *anomaly*. A characteristic challenge in identifying anomalies in such scenarios is the *imbalance* in the distribution of normal and abnormal classes. Most automated processes in industrial manufacturing, and in especially machine-supported tightening operations, run robust and with only narrow margin of variation. Therefore, irregular patterns often occur several orders of magnitude less frequently than recordings of a normal tightening process do [9, 10]. To refer to that underlying dilemma, we consider the problem of *unbalanced class distributions* during the analysis in **Sec. 4**.

2.2. Machine learning methods and time series clustering

Data science and the aforementioned domain of ML enables an automated and intelligent analysis of large amounts of data. For this reason, its methods are well suited for the detection of anomalies in bolting processes [11]. Scientific literature typically distinguishes ML algorithms in three paradigms:

- Supervised learning
- Unsupervised learning
- Reinforcement learning

In *supervised learning*, there is initial knowledge about the class to predict, which in this context is commonly referred to as *label*. *Unsupervised learning* lacks such a label and the models often aim to predict the label for given observations. Lastly, *reinforcement learning* relies on agents to take actions in an environment and maximize a defined reward function [12].

In this paper, we only use unsupervised ML (see **Sec. 3.2**), more precisely, methods for *clustering*. Besides tasks such as *dimension reduction* or *outlier detection*, clustering belongs to one of the main groups of unsupervised ML. Clustering models aim to group a set of objects in such a way that the objects in the same group are more similar to each other than the objects in other groups [7, 13]. We call each group that is formed in this way a *cluster*. This paper aims to cluster tightening data to differentiate between normal from abnormal processes. Since the goal is to perform the clustering for time series, we consider this in the selection of appropriate models. We introduce the selected clustering method and the appropriate distance metric to compare the similarities of time series later in **Sec. 3**.

2.3. On the necessity of an unsupervised approach

In principle, the question arises as to why we seek an application of unsupervised methods when supervised approaches are more widely used in scientific discourse and are already successfully applied [11, 14]. For one, some authors argue that it is too costly to manually record labels for a dataset of tightening processes and to use them as class information for the process analysis [11]. Faults are often only recognized at the end of the line and can only be traced back to the root cause with considerable effort. Additionally, we see another benefit of the unsupervised approach in the potential to identify fault patterns that have not yet occurred. Supervised methods require previously recorded examples of the abnormal class to be identifiable using a respective label. In case of an unseen anomaly, the chances of recognizing it decreases considerably. The unsupervised approach eliminates the need to manually label past observations and allows detecting previously unknown faults.

3. Methods

3.1. Dynamic time warping

The ML methods introduced in **Sec. 3.2** utilize metrics to determine the distance between time series. One of the most common distances is the *Euclidean distance* (ED) that is simply the length of a time segment between two points. *Dynamic Time Warping* (DTW) is another distance metric. The DTW algorithm was originally developed for speech recognition [15] and is nowadays frequently used for time series analysis due to its dynamic shape matching applications [14]. **Fig. 1** shows a comparison of two exemplary time series, x and y, using ED (a) and DTW (b). In contrast to ED, DTW is able to measure a similarity of two time series that vary in speed. DTW is even able to compare series of different length, as indicated in plot (a) by the red points of x that do not have a match in y [16].

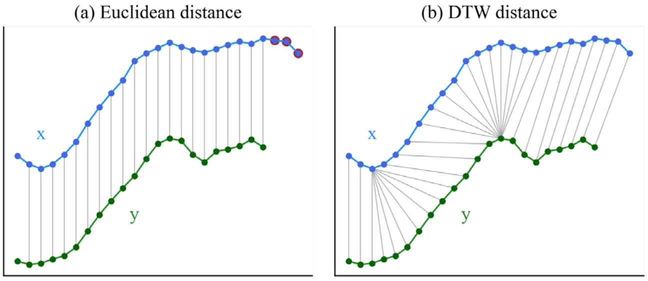


Fig. 1. Comparison of distance metrics for (a) an Euclidean example and (b) an example using Dynamic time warping.

DTW assigns each point of time series x to a point of a second time series y and vice versa [17]. In a special scenario where DTW assigns all points one-to-one, the DTW distance corresponds to ED [18]. To determine DTW, the time series x and y are arranged in a $l_X \times l_Y$ matrix. In this matrix, the element with coordinates (i, j) corresponds to the squared distance d calculated as $\mathrm{d(i,j)} = (\mathrm{x_i} - \mathrm{y_i})^2$. Then, a path $W = w1, w2, \dots, wK$ with the k-th element $w_k = (i, j)_k$ is searched for $max(l_X, l_Y) \leq K < l_X + l_Y - 1$ over which the cumulative distance $\gamma(i, j)$ is minimized. The elements w_k each correspond to the assignment $(i, j)_k$ [19]. Finally, the optimal path is determined by **Eq. 1** [20].

$$DTW(x, y) = min\left\{\sqrt{\sum_{k=1}^{K} w_k}\right. \tag{1}$$

$$\gamma(x, y) = d(x_i, y_i) + min\begin{cases} (y(i-1, j-1) \\ \quad y(i-1, j) \\ \quad y(i, j-1) \end{cases} \tag{2}$$

The determination of the optimal path can then be solved by using dynamic programming and **Eq. 2** [19].

3.2. K-Means clustering

In this paper, the unsupervised ML method k-Means is used. The k-Means algorithm is considered one of the most successful and popular methods for clustering analysis. As such, it is an ML case without the need for labels. K-Means is widely used and has applications across various domains and fields [21].

Let $X = \{x1, \dots, xn\}$ be a dataset in a d-dimensional Euclidean space, $\mathbb{R}^d$. Then $A = \{a_1, \dots, a_c\}$ represent the centers of the c clusters. Using $Z = [Z_{ik}]_{n \times c}$, a cluster membership can be specified, where a binary variable z_{ik} (e.g., $Z_{ik} \in \{0,1\}$) indicates whether the observation x_i belongs to the k-th cluster for $k = 1, \dots, c$. The k-Means algorithm iterates the conditions to minimize the objective function $J(Z, A)$ by redetermining the cluster centers and memberships according to the following two conditions of **Eq. 4** and **Eq. 5** [13].

$$a_k = \sum_{i=1}^{n} z_{ik} x_{ij} \ / \ \sum_{i=1}^{n} z_{ik} \tag{3}$$

$$z_{ik} = \begin{cases} 1, if \ \|x_i - a_k\|^2 = min(\|x_i - a_k\|^2), 1 \leq k \leq c \\ \qquad\qquad else \ 0 \end{cases} \tag{4}$$

To illustrate, we show the clustering with the k-Means algorithm schematically in **Fig. 2**. We illustrate the algorithm in three simple steps (a) to (c) and consider the model for two clusters. Initially, the data contains no knowledge about any class associations (a). As such, k-Means assumes two points as centers of the future classes, which we did randomly in the figure. The green and blue stars constitute the selected *centroids* in this example (b). By using a distance measure, the class of all observations can be determined based on the distance to the new cluster centers. After this assignment, recalculating the cluster centers allows updating the location of the arbitrarily chosen centers. K-Means iterated this step and recalculated the centroids until there are no further changes in the cluster assignments (c). Then, the k-Means clustering is completed [21].

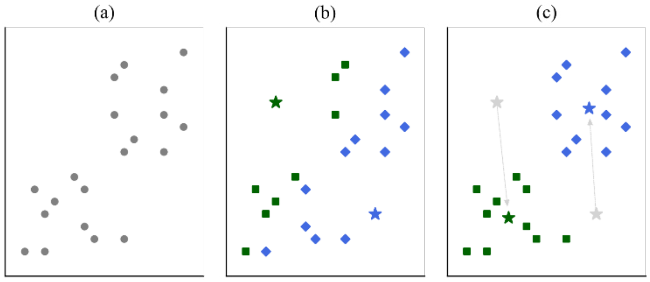


Fig. 2. (a) Example of a k-Means clustering in three steps (a-c).

3.3. Confusion matrix and classification metrics

Analysts often express the results of a classification with the help of *confusion matrices*. A confusion matrix compares the actual classes in a dataset with the predicted classes of a model. In scenarios with just two classes, the observations of a targeted class are often designated as being positive (1), whereas the observations of the other class are designated as negative (0). As such, a confusion matrix has a specific table layout, where each row of the matrix represents instances of the actual class while each column represents instances of the predicted class [22]. **Fig. 3.** illustrates an example of a confusion matrix.

		Predicted values	
		Positive class (1)	Negative class (0)
Actual values	Negative class (0)	True positive (TP)	False negative (FN)
	Positive class (1)	False positive (FP)	True negative (TN)

Fig. 3. Example of a confusion matrix.

In practice, the 2 × 2 matrix holds the actual number of observations for the four values: TP, TN, FN, and FP. TP and TF denote the observations that the classification correctly predicted to be positive and negative, respectively. Similarly, FN refers to the observations that the prediction model incorrectly predicted to be negative, while FP corresponds to the ones that were predicted incorrectly. Then, the *accuracy*, calculated using **Eq. 6**, provides a simple metric to evaluate classifications. Additionally, *precision* and *recall* provide more information about the quality of a classification (see **Eq. 6** and **Eq. 7**).

$$accuracy = \frac{TP+TN}{TP+FN+FP+TN} \tag{5}$$

$$precision = \frac{TP}{TP+FP} \tag{6}$$

$$recall = \frac{TP}{TP+FN,} \tag{7}$$

However, in scenarios with unequally distributed number of classes, these classification metrics are likely to be misleading. For this purpose, the *F1-score* provides a useful metric that takes into account class distributions. **Eq. 8** describes how to calculate it using the harmonic mean of precision and recall.

$$F_1\text{-}score = 2 * \frac{Precision*Recall}{Precision+Recall} = \frac{TP}{TP+\frac{1}{2}(FP+FN)} \tag{8}$$

To treat all classes equally, the analysis will discuss the *macro-averages F1-score*. This score is computed using the arithmetic mean of all F1-scores of every class (see **Eq. 9**).

$$Macro\text{-}average\ F_1\ score = \frac{sum(F1\text{-}scores)}{number\ of\ classes} \tag{9}$$

4. Results

4.1. Objective and approach for time series clustering

We apply the methods presented in **Sec. 3** to find abnormal time series in the tightening data of an automotive manufacturer. We deliberately treat the use case for this as an unsupervised scenario despite the availability of class labels. The goal is to separate the screw runs based on their specific tightening shape. This objective is based on the hypothesis that defective tightening data will have a different appearance compared to normal runs, and can thus be placed in distinct clusters. Based on the number of observations in each cluster, we determine the predicted class using the following pair of rules:

- If the number of observations of a single cluster is *equal* or *above* the total number of observations divided by the number of clusters, we consider all observation of the respective cluster to be of the majority class; in that case to be *normal.*
- If the number of observations of a single cluster is *below* said quotient, we assign the observation to the minority class; in that case to be *abnormal* or faulty.

4.2. Screw driving use case from the automotive industry

The example used in this paper looks at tightening data from a German automotive company. For this case, a comprehensive dataset with automatically recorded data of bolted joints from the engine assembly is available. In order to obtain a clear separation of the classes, we consider a single connection that is prone to failure in practice for just one engine type. Due to the costly, retrospective identification, the number of faulty observations is only 96. The knowledge about these errors is the result of a manual investigation in the end-of-line testing. We traced these anomalies back to the tightening process under consideration. By comparison, we have hundreds of thousands of correct observations that we may use for the analysis. In the remaining paper, we refer to the normal samples as *OK* (‘okay’) and to the abnormal samples as *nOK* (‘not okay’). Furthermore, we treat OK as class 1 and nOK as class 0.

To obtain robust results, we generated ten datasets by merging 5,000 OK samples each with all 96 available nOK samples. As is common during ML model application, we then made a random division into training and testing data using a 70:30 ratio. **Table 1** summarizes the distribution of classes. Decreasing the ratio of unequally distributed OK and nOK classes reduced the required duration for modeling, but also kept the imbalanced nature of the scenario in place to challenge the approach.

Table 1. Distribution of OK- and nOK-classes in training and test data of the ten generated datasets.

Dataset number	Training data		Test data	
	OK	nOK	OK	nOK
0	3,499	68	1,501	28
1	3,499	68	1,501	28
2	3,501	66	1,499	30
3	3,492	75	1,508	21
4	3,501	66	1,499	30
5	3,504	63	1,496	33
6	3,498	69	1,502	27
7	3,501	66	1,499	30
8	3,508	59	1,492	37
9	3,506	61	1,494	35

As such, the reduction of the dataset class ratio does not reduce the validity of the approach. Instead, by later calculating averages of the classification scores for ten models, we manage to smooth out random outliers and get a more robust modelling result. To provide a visual sense of the tightening data, **Fig. 4** shows 25 randomly selected OK- and nOK-observations.

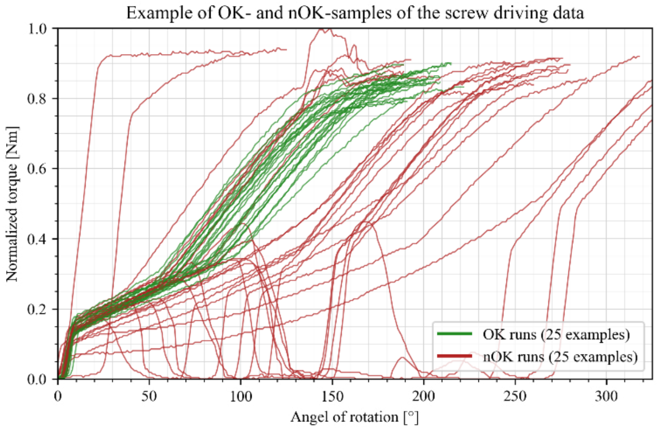


Fig. 4. Exemplary display of 25 OK and 25 nOK screw runs.

4.3. Implementation of k-means for tightening data

We deploy the k-Means model using the Python (v3.7.9) package `tslearn` (v0.5.0.5) [23]. Tslearn provides different ML tools for the analysis of time series. The package builds and depends on the libraries `scikit-learn` (v0.24.1), `numpy` (v1.19.5) and `scipy` (v1.6.1). In addition to other implementations, such as a kernel k-Means or k-shape clustering, tslearn provides `TimeSeriesKMeans`, a model for implementing k-Means clustering for time series [23]. As a distance measure to determine the similarity of observations, we deploy the DTW metric introduced during **Sec. 3**. The full analysis is publicly available on GitHub [24]. However, due to data protection considerations, it was not possible to publish the tightening data. Lastly, since there is no inherently correct amount of clusters to generate, we vary the number of k-Means clusters between 2 and 10. Hence, in total, we train and evaluate 90 models.

4.4. Results of the clustering-based anomaly detection

For reference, **Appendix A** shows the full set of parameters of the *k*-means model. After performing the analyses, we evaluate the results. First, we calculate the scores for TP, FP, FN, and TN of all of the cluster analyses. To jointly display the results, we determine relative shares for TP and FP as well as for FN and TN. Furthermore, since there is a total of ten datasets for each number of clusters (ranging from two clusters up to a total of ten) there are also ten clustering results each. We determine the mean value for each score across the different numbers of clusters. **Fig. 5** displays the averaged and normalized values of the four scores. High values for TP and TN correspond to good model results.

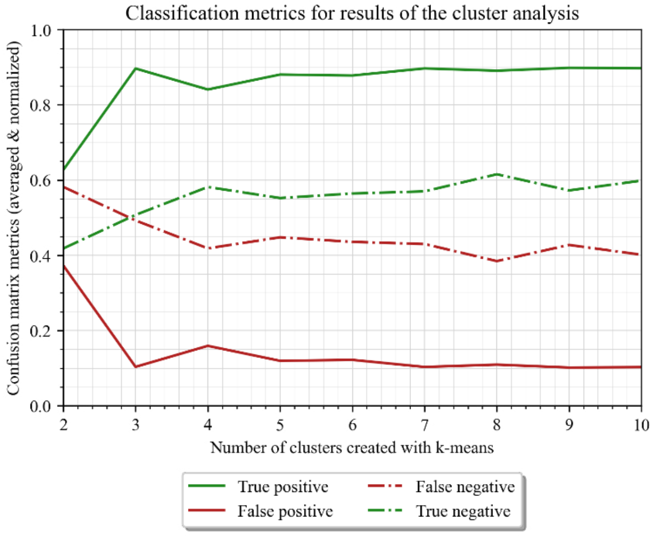


Fig. 5. TP, FN, FP and TN for ten clusters (and averaged for ten datasets).

The figure shows that the scores of TP and TN marked in green are in most cases above the scores of FN and FP marked in red. This means that with the help of the approach, starting at a cluster number of more than two, significantly more observations were correctly identified as normal (TP) than as faulty (FP). At the same time, starting at a cluster number of more than three, the number of bolted joints correctly identified as abnormal (TN) is continuously higher than the number of observations that are falsely identified as being correct (FN).

Table 2. Summary of analysis results with additional classification metrics.

Number of clusters	Accuracy	Precision	Recall	Macro-avg. F1 score
2	62,35 ± 22,62	62,70 ± 23,58	77,48 ± 12,42	49,18 ± 13,92
3	88,89 ± 1,66	89,65 ± 1,85	98,97 ± 0,00	63,65 ± 1,91
4	83,56 ± 1,28	84,07 ± 1,37	98,99 ± 0,00	53,54 ± 0,74
5	87,35 ± 0,99	88,05 ± 1,11	98,99 ± 0,00	58,44 ± 1,24
6	87,15 ± 0,61	87,79 ± 0,67	99,02 ± 0,00	56,24 ± 0,64
7	89,02 ± 0,37	89,68 ± 0,41	99,04 ± 0,00	57,94 ± 0,62
8	88,53 ± 0,35	89,07 ± 0,37	99,14 ± 0,00	56,94 ± 0,29
9	89,19 ± 0,34	89,83 ± 0,36	99,06 ± 0,00	56,91 ± 0,29
10	89,14 ± 0,38	89,73 ± 0,41	99,11 ± 0,00	57,92 ± 0,47

Besides accuracy, precision and recall, **Table 2** shows the macro-avg. F1-scores, averaged for the ten datasets. Since the unequally distributed classes do not influence that score, it better shows the performance of the analysis. Except for a cluster number of 2, the score is above 50% for all cases and thus above random speculation. In the best case, when we generated three clusters, the macro-avg. F1-score is highest at 63.65%.

Fig. 6 shows an exemplary clustering result for dataset zero and for a clustering with five groups. While clusters 1, 3 and 4 contain mainly OK observations, clusters 2 and 5 contain exclusively nOK samples. This example illustrates the advantage of the methodology. If the pattern of tightening data differs sufficiently, such distinct observations fall into relatively small clusters. Application of the rule set from **Sec. 4.1** then flags these observations as faulty tightening runs. With that, the tightening anomalies emerge as a novel and distinct cluster.

5. Conclusion

In summary, the presented approach helps to find anomalies in tightening data. We consider the real-world application from the automotive industry a success, since we were able to detect distinct anomalies. However, it is required that abnormal tightening observations differ clearly in their pattern compared to the normal observations. Thus, the unsupervised approach is not the only (or the easiest) way to find these particular observations. Nevertheless, a major advantage of the approach is that it did not require extensive investigation of past tightening errors to detect anomalies prior to the ML analysis.

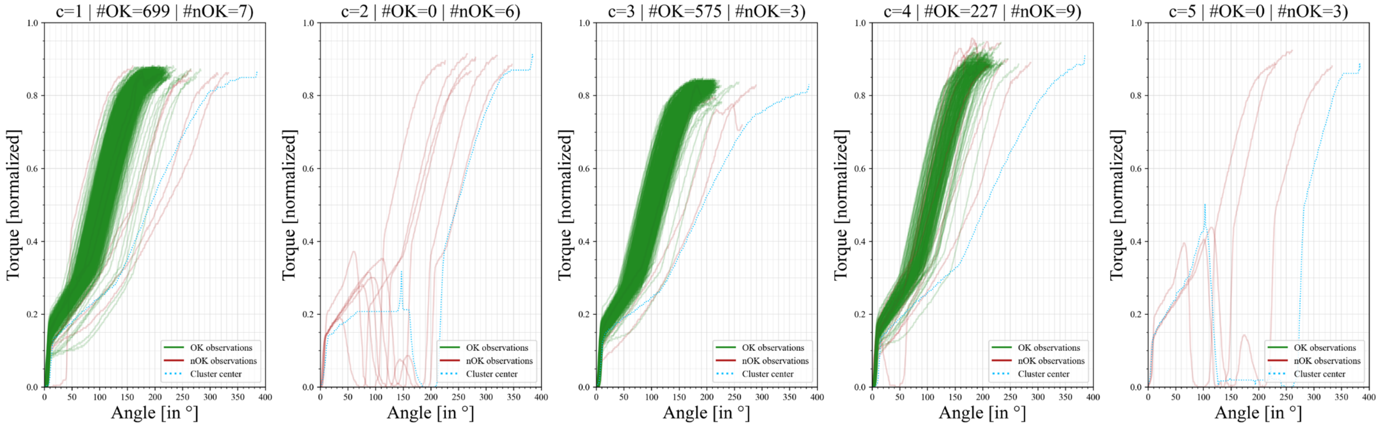


Fig. 6. Visualization of the results for a cluster analysis with a DTW-based *k*-means for five clusters (exemplary results for dataset #0)

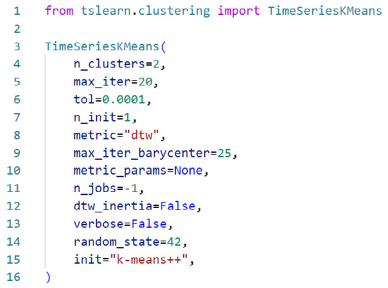
Furthermore, the approach was not able to detect all anomalies. For example, cluster 4 in **Fig. 6** shows a different type of anomalous deviation in the final tightening phase. Despite this, the DTW-based clustering approach has placed the observations in a group dominated mainly by OK observations. We believe it is necessary to develop further unsupervised approaches to detect tightening data anomalies, even if these errors occur extremely rarely or just once.

In future work, we will continue to focus on unsupervised anomaly detection. A particular focus of the activities is the detection of anomalies within value chain networks. We aim to develop an anomaly detection approach that works across value chains and aids in identifying multivariate anomalies.

Acknowledgements

This paper received partial funding by the project "Cross-process chain detection of material and process anomalies at imbalanced data for technical plastic assemblies" (ProData) from the German ministry of education and research (BMBF) as part of funding program for "Strengthening the data skills of the next generation of scientists" (16-DKWN-119A).

Appendix A. Code excerpt with model parameters



```python
from tslearn.clustering import TimeSeriesKMeans

TimeSeriesKMeans(
    n_clusters=2,
    max_iter=20,
    tol=0.0001,
    n_init=1,
    metric="dtw",
    max_iter_barycenter=25,
    metric_params=None,
    n_jobs=-1,
    dtw_inertia=False,
    verbose=False,
    random_state=42,
    init="k-means++",
)
```

References

[1] J. Lee, H. D. Ardakani, S. Yang, and B. Bagheri, "Industrial big data analytics and cyber-physical systems for future maintenance & service innovation," *Procedia CIRP*, vol. 38, no. 1, pp. 3–7, 2015, doi: 10.1016/j.procir.2015.08.026.

[2] R. Wöstmann, A. Barthelmey, N. West, and J. Deuse, "A retrofit approach for predictive maintenance," *Proceedings MHI*, vol. 4, no. 1, pp. 94–106, 2019, doi: 10.1007/978-3-662-59317-2_10.

[3] J. Deuse, N. West, and M. Syberg, "Rediscovering scientific management: The evolution of industrial engineering to industrial data science," *International Journal of Production Management and Engineering*, vol. 10, no. 1, pp. 1–12, 2022, doi: 10.4995/ijpme.2022.16617.

[4] N. West, T. Schlegl, and J. Deuse, "Feature extraction for time series classification using univariate descriptive statistics and dynamic time warping in a manufacturing environment," *IEEE ICBAIE*, vol. 2, no. 1, pp. 762–768, 2021, doi: 10.1109/ICBAIE52039.2021.9389954.

[5] W. A. Shewhart, *Economic Control of Quality of Manufactured Product*. Milwaukee, WI, USA: American Society for Quality Control, 1931.

[6] R. Chumakov, "An artificial neural network for fault detection in the assembly of thread-formed screws," *JIM*, vol. 19, no. 1, pp. 327–333, 2008, doi: 10.1007/s10845-008-0085-5.

[7] C. A. Ratanamahatana, E. Keogh, A. Bagnall, and S. Lonardi, "A novel bit level time series representation with implications for similarity search and clustering," *Proceedings PAKDD*, no. 1, pp. 771–777, 2005, doi: 10.1007/11430919_90.

[8] Atlas Copco, *Pocket guide to screwdriving*. Stockholm, Sweden: Atlas Copco Industrial Technique AB, 2016.

[9] T. Schlegl, S. Schlegl, D. Tomaselli, N. West, and J. Deuse, "Adaptive similarity search for the retrieval of rare events from large time series databases," *Advanced Engineering Informatics*, vol. 52, no. 101629, pp. 1–12, 2022, doi: 10.1016/j.aei.2022.101629.

[10] T. Schlegl, S. Schlegl, A. Sciberras, N. West, and J. Deuse, "Margin-based greedy shapelet search for robust time series classification of imbalanced data," *IEEE International Conference on Big Data*, no. 1, pp. 5266–5274, 2021, doi: 10.1109/BigData52589.2021.9671952.

[11] D. Riberio, L. M. Matos, P. Cortez, G. Moreira, and A. Pilastri, "A comparison of anomaly detection methods for industrial screw tightening," *Computational Science and Its Applications*, vol. 12950, no. 1, pp. 1–16, 2021, doi: 10.1007/978-3-030-86960-1_34.

[12] F. Emmert-Streib and M. Dehmer, "Taxonomy of machine learning paradigms: A data-centric perspective," *WIREs DMKD*, vol. 12, no. 5, pp. 1–24, 2022, doi: 10.1002/widm.1470.

[13] K. P. Sinaga and M.-S. Yang, "Unsupervised k-means clustering algorithm," *IEEE Access*, vol. 8, no. 1, pp. 80716–80727, 2020, doi: 10.1109/ACCESS.2020.2988796.

[14] T. Fu, "A review on time series data mining," *EAAI*, vol. 24, no. 1, pp. 164–181, 2011, doi: 10.1016/j.engappai.2010.09.007.

[15] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE TASSP*, vol. 26, no. 1, pp. 43–49, 1978, doi: 10.1109/TASSP.1978.1163055.

[16] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: A Survey," *IEEE TKDDA*, vol. 26, no. 9, pp. 2250–2267, 2014, doi: 10.1109/TKDE.2013.184.

[17] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data: Experimental Comparison of Representations and Distance Measures," *VLDB Endowment*, vol. 2, no. 1, pp. 1542–1552, 2008, doi: 10.14778/1454159.1454226.

[18] T. Rakthanmanon *et al.*, "Searching and mining trillions of time series subsequences under dynamic time warping," *ACM SIGKDD*, vol. 18, no. 1, pp. 262–270, 2012, doi: 10.1145/2339530.2339576.

[19] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," *Proceedings ICKDDM*, vol. 3, no. 1, pp. 359–370, 1994, doi: 10.5555/3000850.3000887.

[20] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time wapring," *KIS*, vol. 7, no. 3, pp. 358–386, 2004, doi: 10.1007/s10115-004-0154-9.

[21] M. Ahmed, R. Seraj, and S. M. Shamsul Islam, "The k-means algorithm: A comprehensive survey and performance evaluation," *MDPI EJ*, vol. 9, no. 8, pp. 1–12, 2020, doi: 10.3390/electronics9081295.

[22] M. J. Zaki and W. Meira Jr., *Data mining and analysis: Fundamental concepts and algorithms*. New York: Cambridge Univ. Press, 2014.

[23] R. Tavenard *et al.*, "Tslearn: A machine learning toolkit for time series data," *JMLR*, vol. 21, no. 118, pp. 1–6, 2020, link: jmlr.org/papers/v21/20-091.html

[24] N. West, *Project repository*, 2023, link: github.com/nikolaiwest/2022-anomaly-detection-cirp