A Nearest Neighbor-Based Concept Drift Detection Strategy for Reliable Condition Monitoring

Nicolas Jourdan
Institute for Production Management, Technology and Machine Tools
Technical University of Darmstadt, Germany
nicolas.jourdan@tu-darmstadt.de

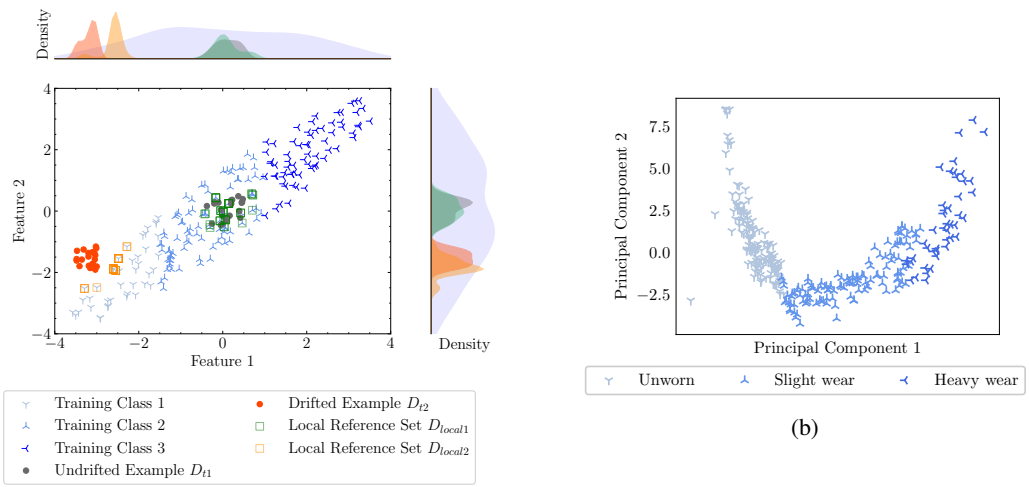
Abstract

Condition monitoring is one of the most prominent industrial use cases for machine learning today. As condition monitoring applications are commonly developed using static training datasets, their long-term performance is vulnerable to concept drift in the form of time-dependent changes in environmental and operating conditions as well as data quality problems or sensor drift. When the data distribution changes, machine learning models can fail catastrophically. We show that two-sample tests of homogeneity, which form the basis of most of the available concept drift detection strategies, fail in this domain, as the live data is highly correlated and does not follow the assumption of being independent and identically distributed (i.i.d.) that is often made in academia. We propose a novel drift detection approach called Localized Reference Drift Detection (LRDD) to address this challenge by refining the reference set for the two-sample tests. We demonstrate the performance of the proposed approach in a preliminary evaluation on a tool condition monitoring case study.

1 Introduction

In modern manufacturing, condition monitoring plays an increasing role in ensuring equipment availability, reducing defects, and optimizing production efficiency. Within this domain, machine learning (ML) has contributed significantly to recent technological developments [13, 5]. ML models can analyze sensor data and accurately predict the condition or wear state of equipment and tools, thereby reducing the need for manual inspections and the likelihood of unplanned downtimes. As ML applications transcend from academic research to real-world usage, questions regarding their continuous reliability and robustness arise. Production-grade models need to be able to handle uncertainties such as degrading equipment or data quality problems from streaming failures or faulty sensors [7]. The training dataset of an ML model is restricted to a certain state of the manufacturing process in time. After model deployment though, the manufacturing environment will likely encounter changes such as aging sensors, data quality deviations or changes in the factory layout and machine placement that are not captured in the training dataset, a scenario referred to as *concept drift* or *distribution shift*. In the context of this publication, we define concept drift as $P_{train}(\boldsymbol{x}, y) \neq P_{online,t}(\boldsymbol{x}, y)$, where P_{train} and $P_{online,t}$ denote the joint distributions of input data $\boldsymbol{x}$ and target data y during training and deployed usage of the model at time t, respectively. Concept drift can lead to reduced performance during the operation of an ML model in contrast to the performance that was evaluated on a static test dataset during development, up to the point where the ML model fails catastrophically. It is therefore important to detect when drift occurs and raise alarms accordingly to guarantee a continuously reliable model [2, 6]. In case of a detected drift, the root cause needs to be identified and the ML model potentially retrained using more recent data. As this issue is relevant throughout almost all application domains of ML, various methods for drift detection have been proposed in the literature. On a high level, drift detection methods commonly require a

Workshop on Distribution Shifts, 37th Conference on Neural Information Processing Systems (NeurIPS 2023).



(a) (b)

Figure 1: **Left:** Visualization of the LRDD mechanism with synthetic data and an in-distribution deployment data example as well as a drifted deployment data example. Conventional two-sample tests that compare the deployment window distribution to the complete training distribution would wrongly indicate drift in both cases. Comparison with the localized reference distribution shows homogeneity for the in-distribution example, but correctly indicates drift for the drifted example. **Right:** 2D-PCA plot of the features in the PHM 2010 dataset for all three wear classes.

reference/training dataset as well as a set of deployment/online samples. These sets can be formed, among other options, simply by the input features, model-dependent transformations of the features or model error rates if labels are available [9]. The sets are typically compared using a two-sample hypothesis test for a chosen measure of distributional equality which alerts the user to drifts at a given significance level α [2]. However, this is done under the assumption of the samples in both sets are independent and identically distributed (i.i.d.). In a typical tool condition monitoring (TCM) application, this assumption does not hold as the data points that are collected over time are highly correlated. Thus, if only a certain window in time is used to assess concept drift, we hypothesize that conventional drift detection methods will yield a high number of false positive detections as they compare deployment time windows that only contain a certain subset of tool conditions with the full reference dataset, containing all conditions. This systemic issue is recognized in a recent publication by Cobb et al. [2], that proposes *Context-Aware Drift Detection*. In their approach, the authors extend a multivariate two-sample testing approach that uses Maximum Mean Discrepancy (MMD) [4] by a context variable. This context variable can be, among others, the classifier prediction or the time of day, and is consequently used to weigh the reference set and increase the relevance of data that corresponds to a context that is close to the current deployment context.

We hypothesize, that even if the classifier prediction class is used as context, the intra-class variance will still yield false positives in the TCM use case. We thus propose a simple and computationally cheap alternative, that we call Localized Reference Drift Detection (LRDD). By choosing an adaptive local reference set to the current deployment data through nearest neighbor search, we aim to make false alarms for concept drift less likely without the need to explicitly define a context variable. To summarize, the contributions of this paper are: **(1)** A simple, yet effective solution to concept drift detection in practical scenarios that do not follow the i.i.d. assumption and **(2)** a preliminary case study of the proposed approach with an industrial tool condition monitoring dataset.

2 Methodology

The LRDD algorithm pseudocode is given in Algorithm 1 and visualized in Figure 1 (left). It consists of the following procedure: At training time of the ML model, a nearest neighbor model is fitted to the test split D_{ref} of the available dataset. Now, during the operation of the ML model at time t, a data window $D_t = \{x, \hat{y}\}$ that contains the n most recent input data points x as well as the

Algorithm 1 Localized Reference Drift Detection (LRDD)

Require: $D_{\text{ref}}, D_t, k, \alpha, N_{\text{perm}}$
Ensure: Drift detection result
1: Initialize k-NN with D_{ref} to get NN; Initialize $I_{NN} = \emptyset$
2: **for** $(x_i, \hat{y}_i)$ in D_t **do**
3: Get k nearest neighbors of x_i in D_{ref} with same label as $\hat{y}_i$; Update I_{NN}
4: Extract D_{local} from D_{ref} using unique indices in I_{NN}; Compute MMD^2 between D_t and D_{local}
5: Compute p-value using MMD^2 and N_{perm} permutations
6: **if** $p < \alpha$ **then return** Drift detected
7: **elsereturn** No drift detected

corresponding model predictions $\hat{y}$ up until time t is used to detect concept drift. Therefore, the k-nearest neighbors of the data points in D_t are queried from the nearest neighbor model, only considering data points in D_{ref} that have class labels y that correspond to the ML models estimates $\hat{y}$. Following this step duplicates in the resulting k-neighbor data points are removed. The resulting set of unique k-neighbors forms the localized reference set D_{local}, which can be compared against D_t to check for concept drift. This approach allows a refined comparison of the current data distribution with a sub-distribution of the reference set that is closest in feature space, therefore considering the correlation between the data points in D_t and preventing false positive detections when D_t is not i.i.d. and does not contain samples from all classes or feature regions in D_{ref}. In case of drift, we expect the samples in D_{local} to be far away in feature space from the samples in D_t which can be detected through two-sample testing. The two-sample test can be performed with any existing method for checking distributional equality such as permutation testing with MMD or a combination of univariate Kolmogorov Smirnov-tests for all feature dimensions [9].If the chosen two-sample test yields a p-value that is below the chosen significance level α, concept drift is indicated which means that the current ML model may be unreliable for the incoming data. For the drift detection in LRDD, we resort to the kernel-based MMD method for multivariate two-sample testing [4] that is part of other drift detectors as well [2, 9]. The MMD is defined as a distance measure between two distributions P and Q based on the mean embeddings μ_p and μ_q in a reproducing kernel Hilbert space F: $MMD(F, P, Q) = \|\mu_p - \mu_q\|_F^2$. A p-value for testing distributional equality between p and q is obtained via permutation testing. While this method is popular in literature, other two-sample tests can be used in combination with our methodology as well.

3 Case Study

For the case study, we utilize a tool condition monitoring dataset that was released in the scope of the 2010 Prognostics and Health Management (PHM) challenge [10] and was used in a number of recent TCM publications [11, 14]. The dataset contains the data of seven measured sensor signals as time series recorded at 50 KHz during a CNC milling process: cutting forces [N] (X, Y, Z), accelerations [g] (X, Y, Z) and the root mean square of the signal [V] of an acoustic emission sensor (AE). The datasets contains six sets that correspond to the cutters used in the experiments. Each of the six sets contains 315 cuts. For three of the sets (c1, c4, c6), labels exist for each cut in the form of flank wear measurements of the individual flutes that were done using a microscope. We thus only consider c1, c4 and c6 in our experiments. The cuts were made at a spindle speed of 10400 rpm with a feed rate of 1555 $\frac{\text{mm}}{\text{min}}$. Following [14], we extract 8 features from all of the signals individually, including both statistical and spectral indicators. For the condition monitoring case study, we generate three classification labels from the maximum flute flank wear VB_{max} of the individual cuts: (1) *Unworn* for $\text{VB}_{max} <= 70$, (2) *Slight wear* for $130 >= \text{VB}_{max} > 70$ and (3) *Heavy wear* for $\text{VB}_{max} > 130$. The class limits were chosen to build subsets of approximately equal size. We split the three cutter-specific sub-datasets into a training dataset (2 cutters) and a test dataset (1 cutter) and train a random forest classification model on the extracted and standardized features. A 2-dimensional PCA plot of the extracted features for one of the cutter sub-datasets (c1) is visualized in Figure 1 (right). The evaluation results of the condition monitoring performance are stated in Table 1.

For evaluating the drift detection performance on this dataset, we split the testing data used for classifier evaluation, into two equally-sized subsets for all three permutations, *cf.* Table 1. One forms the reference dataset D_{ref}, while the other one serves as the simulated deployment data. We

Table 1: Accuracy of the condition monitoring application.

Training Set	Testing Set	Accuracy
c1, c4	c6	0.8
c4, c6	c1	0.87
c1, c6	c4	0.84

Table 2: Drift detection results for LRDD and selected baselines.

	Precision	Recall	F1
MMDDrift	0.54	1.00	0.70
KSDrift	0.58	1.00	0.73
ContextMMDDrift	0.70	1.00	0.82
LRDD	0.97	1.00	0.98

randomly sample 100 intervals of size $n = 30$ from the deployment data as the current data windows D_t. To simulate unseen process conditions, we randomly exclude one of the three classes from the reference dataset. This class is then treated as *drifted* as it is not included in the training dataset of the classifier. If the samples in the deployment window D_t fall into this *drifted* process state, the detectors should indicate drift, while no drift should be detected for the *undrifted* other two classes as they are present in the reference dataset. The ratio of *drifted* vs. *undrifted* samples is set to 50%.

We compare the performance of the proposed LRDD approach with three common baseline drift detectors: **MMDDrift**: Kernel-based maximum mean discrepancy using permutation testing [4, 9]. **KSDrift**: Two-sample testing using univariate Kolmogorov-Smirnov tests on all feature dimensions with additional Bonferroni-correction [1]. **ContextMMDDrift**: Similar to MMDDrift but with additional context variable [2] as briefly explained in the introduction. We utilize the classifier class predictions as the context variable. For the baseline drift detectors we use the implementations in the *alibi-detect* [12] python package with *pytorch* backend. We use $\alpha = 0.05$ and $N_{perm} = 100$ in all experiments. For LRDD we set $k = 5$.

Drift detection results are reported in Table 2. Notably, all utilize detectors have a recall of 100%, indicating that all drifts are reliably detected. The main improvement through the proposed methodology lies in the increased precision, which is significantly higher than that of all baseline detectors. This confirms two hypotheses: First, conventional drift detectors that compare the full reference dataset with deployment data that is non-i.i.d. will yield a high number of false positives. Second, the proposed methodology for selecting a localized reference set enables highly accurate drift detection even if the deployment samples have a high correlation through the sample acquisition time. In the scope of the TCM case study, the results show that LRDD can be used to indicate the reliability of the condition monitoring model over its usage time to operators. In case of a detected drift, operators should perform a root cause analysis to investigate potential data quality problems and decide whether the model should be retrained with more recent data.

4 Conclusion

In this paper we proposed a novel method for concept drift detection tailored to applications like condition monitoring, where the i.i.d. assumption for deployment data cannot be made. By focusing on more localized comparisons, the proposed LRDD method aims to strike a balance between sensitivity and specificity. We demonstrated the effectiveness of the proposed method in a preliminary tool condition monitoring case study, highlighting its value for industrial applications. As the size of the utilized dataset is limited, a larger-scale evaluation on suitable datasets is required. For this purpose, a new dataset containing explicitly drifted data is currently generated at a testing machine in a laboratory setup. Furthermore, the influence of the number of neighbors k, the size of the deployment data window D_t as well as finding a suitable α for a given dataset are open issues for systematic investigation. In addition, the impact of using deep feature representations of neural networks for the two-sample tests instead of the unprocessed input data should be investigated, as this has shown to be effective in general out-of-distribution detection tasks [8, 3].

References

[1] J Martin Bland and Douglas G Altman. Multiple significance tests: the bonferroni method. *British Medical Journal*, 310(6973):170, 1995.

[2] Oliver Cobb and Arnaud Van Looveren. Context-aware drift detection. In *International Conference on Machine Learning*, pages 4087–4111. PMLR, 2022.

[3] Silvio Galesso, Max Argus, and Thomas Brox. Far away in the deep space: Nearest-neighbor-based dense out-of-distribution detection. *arXiv preprint arXiv:2211.06660*, 2022.

[4] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[5] Nicolas Jourdan, Lukas Longard, Tobias Biegel, and Joachim Metternich. Machine learning for intelligent maintenance and quality control: A review of existing datasets and corresponding use cases. Hannover : publish-Ing., 2021.

[6] Nicolas Jourdan, Sagar Sen, Erik Johannes Husom, Enrique Garcia-Ceja, Tobias Biegel, and Joachim Metternich. On the reliability of machine learning applications in manufacturing environments. *NeurIPS Workshop on Distribution Shifts*, Dec 2021.

[7] Andrew Kusiak. Smart manufacturing must embrace big data. *Nature*, (544), 2017.

[8] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018.

[9] Stephan Rabanser, Stephan Günnemann, and Zachary Lipton. Failing loudly: An empirical study of methods for detecting dataset shift. *Advances in Neural Information Processing Systems*, 32, 2019.

[10] PHM Society, May 2010.

[11] Mathias Van Herreweghe, Mathias Verbeke, Wannes Meert, and Tom Jacobs. A machine learning-based approach for predicting tool wear in industrial milling processes. In *Machine Learning and Knowledge Discovery in Databases: International Workshops of ECML PKDD 2019, Würzburg, Germany, September 16–20, 2019, Proceedings, Part II*, pages 414–425. Springer, 2020.

[12] Arnaud Van Looveren, Janis Klaise, Giovanni Vacanti, Oliver Cobb, Ashley Scillitoe, Robert Samoilescu, and Alex Athorne. Alibi detect: Algorithms for outlier, adversarial and drift detection, 2019.

[13] Thorsten Wuest, Daniel Weimer, Christopher Irgens, and Klaus-Dieter Thoben. Machine learning in manufacturing: advantages, challenges, and applications. *Production & Manufacturing Research*, 4(1):23–45, 2016.

[14] Yuqing Zhou, Gaofeng Zhi, Wei Chen, Qijia Qian, Dedao He, Bintao Sun, and Weifang Sun. A new tool wear condition monitoring method based on deep learning under small samples. *Measurement*, 189:110622, 2022.