On The Reliability Of Machine Learning Applications In Manufacturing Environments

Nicolas Jourdan
TU Darmstadt, Germany
n.jourdan@ptw.tu-darmstadt.de

Sagar Sen
SINTEF, Norway
sagar.sen@sintef.no

Erik Johannes Husom
SINTEF, Norway
erik.husom@sintef.no

Enrique Garcia-Ceja
SINTEF, Norway
e.g.mx@ieee.org

Tobias Biegel
TU Darmstadt, Germany
t.biegel@ptw.tu-darmstadt.de

Joachim Metternich
TU Darmstadt, Germany
j.metternich@ptw.tu-darmstadt.de

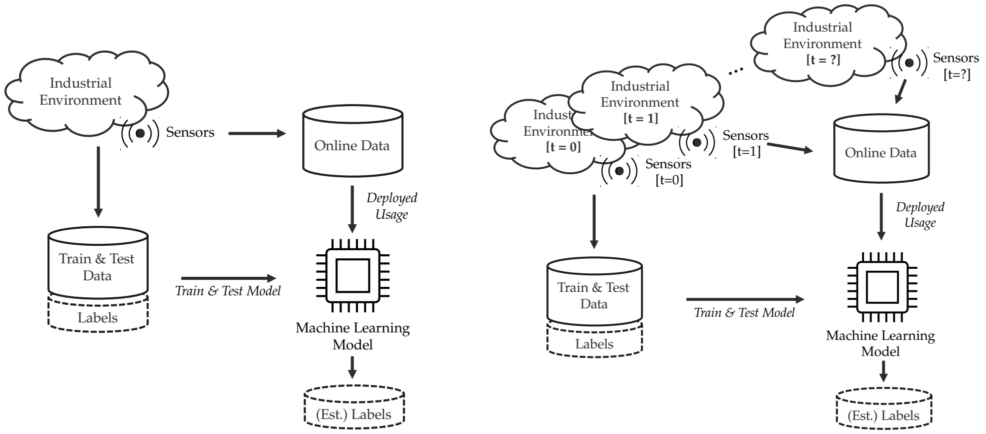
Abstract

The increasing deployment of advanced digital technologies such as Internet of Things (IoT) devices and Cyber-Physical Systems (CPS) in industrial environments is enabling the productive use of machine learning (ML) algorithms in the manufacturing domain. As ML applications transcend from research to productive use in real-world industrial environments, the question of reliability arises. Since the majority of ML models are trained and evaluated on static datasets, continuous online monitoring of their performance is required to build reliable systems. Furthermore, concept and sensor drift can lead to degrading accuracy of the algorithm over time, thus compromising safety, acceptance and economics if undetected and not properly addressed. In this work, we exemplarily highlight the severity of the issue on a publicly available industrial dataset which was recorded over the course of 36 months and explain possible sources of drift. We assess the robustness of ML algorithms commonly used in manufacturing and show, that the accuracy strongly declines with increasing drift for all tested algorithms. We further investigate how uncertainty estimation may be leveraged for online performance estimation as well as drift detection as a first step towards continually learning applications. The results indicate, that ensemble algorithms like random forests show the least decay of confidence calibration under drift.

1 Introduction and Motivation

Increasing digitization and the deployment of advanced technologies in the context of Internet of Things (IoT) and Industry 4.0 are transforming manufacturing lines into Cyber-Physical Systems (CPS) that generate large amounts of data. The availability of this data enables a multitude of applications, including the development and deployment of ML algorithms for use cases such as condition monitoring, predictive maintenance and quality prediction [1]. As ML applications are deployed to productive usage, their continuous reliability has to be guaranteed to protect human operators as well as the financial investments involved. Manufacturing environments are fast changing, highly dynamic and inherently uncertain which poses the requirement for ML applications to be able to adapt to changing environments with reasonable effort and cost [2]. While the ability of adapting to a changing environment is often seen as a default property of machine learning [2], studies show,

Workshop on Distribution Shifts, 35th Conference on Neural Information Processing Systems (NeurIPS 2021).



(a) Stationary environment

(b) Environment under concept and sensor drift

Figure 1: Supervised machine learning in stationary conditions with static sensors (a) in contrast to a dynamic system where the environment as well as the sensors used for perception are non-stationary, which applies to the majority of manufacturing use cases of ML (b). In the latter case, static training/testing sets do not provide continuous performance guarantees. Figure based on [10], adapted and extended with permission of the original authors.

that the generalization ability of a model mainly depends on the configuration and variety of the available training data and is far from guaranteed [3, 4]. Long-term reliability and the handling of uncertainties caused by degrading equipment or faulty sensors are seen as key factors and major hurdles when deploying ML systems in manufacturing environments [5, 6]. With respect to safety certification and risk assessment, online performance monitoring and uncertainty estimation are seen as critical for detecting drifts in the data distribution as well as estimating error magnitudes [6]. In the context of quality management, [7] showed, that the majority of the analyzed frameworks still lack any form of uncertainty estimation or online monitoring.

In this work, we analyze the long-term reliability of ML applications in the manufacturing industry, highlighting the domain-specific issues and potential sources of drift. We benchmark a set of ML algorithms that are most relevant in this domain for robustness to time-dependent drift on an industrial dataset. Further, we assess uncertainty estimation techniques and highlight their potential utility for online performance estimation and drift detection in the context of continual learning to overcome the issue of silently failing ML applications.
Similar experiments have been conducted in [8] but did not consider non-deep learning algorithms, which are of high relevance in the manufacturing domain. Additionally, the introduced drifts were synthetic, while we evaluate on real-world time-dependend drifts. The performance degradation of a classifier (support vector machine) on the utilized dataset has been observed in [9]. In contrast to the existing work, we extend the analysis to a broad spectrum of commonly used ML algorithms and additionally analyze the expressiveness of uncertainty estimation techniques suitable for the respective algorithms.

2 Methodology and Experiments

Concept drift refers to a change in the underlying data distributions of machine learning applications. Especially in the context of pattern recognition, the terms *covariate shift* or *dataset shift* are used interchangeably [11]. Concept drift in the context of this publication, *cf.* Figure 1 (b), can be defined 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 samples $\boldsymbol{X}$ and target labels Y during training and deployed usage of the model at time t, respectively.
Relevant short- and long-term sources of changes in manufacturing environments which may influence the reliability of ML models include: Tool and machine wear [5, 6, 12, 13], changes in product configurations and material properties [12, 13], changes in upstream processes [13], changes in factory layout and machine placement [14], differences in operator preferences and training [12, 13],

seasons and time of day [6], environmental conditions such as temperature or humidity [6, 12, 13], sensor failure/drift/recalibration [5, 6] and data transmission problems [5].
Reliable machine learning applications in dynamic environments may be established in two ways: Either the model and data acquisition setup employed are robust against the relevant sources of drift, e.g. [15], or the model is continually assessed and, if required, adapted to the current environment in a continual learning setup [16].

Commonly, ML models with trained parameters $\boldsymbol{\theta}$ in classification tasks produce probability estimates $p\left(\hat{y}_c \mid \boldsymbol{x}, \boldsymbol{\theta}\right)$ for all classes $c \in \{1, \ldots, C\}$, given a sample of data $\boldsymbol{x}$. The probability may be used to assess the models' confidence/uncertainty in its own prediction. The confidence is referred to as *well-calibrated*, when empirically, it is equal to the probability of the corresponding sample being correctly classified [17]. Thus, confidence estimates that are well-calibrated even in the presence of concept drift, may be used to reason about the reliability of a ML model and determine if it should be adapted, i.e., retrained with new data. Additionally, well-calibrated confidence estimates may be used to identify product configurations or situations that are difficult for the ML model to handle. In the example of quality estimation, this could, e.g., trigger an additional human quality control for a given part or the manual inspection of a machine.

2.1 ML Algorithms and Uncertainty Estimation Methods

To maximize the value for practical applications, we assess ML algorithms that have been identified as most commonly used in the manufacturing domain by a recent review study [18]. We explicitly include non-deep learning algorithms in the analysis, as these are highly relevant in the manufacturing domain, where labeled datasets are often small. In the scope of this work, we focus on classification tasks. Implementation is done using [19] and [20]. The hyperparameters are selected by performing a grid-search over the parameter space of the models, optimizing for accuracy. For each algorithm, we employ a confidence/uncertainty estimation method as described below:

Support Vector Machine (SVM) confidence estimates are obtained using Platt scaling [21] of the sample distances to the separating hyperplane. The parameters for confidence estimation are fitted via 5-fold cross validation.

Decision Tree (DT) confidence estimates are computed as the fraction of training samples of the same class in the leaf node [19].

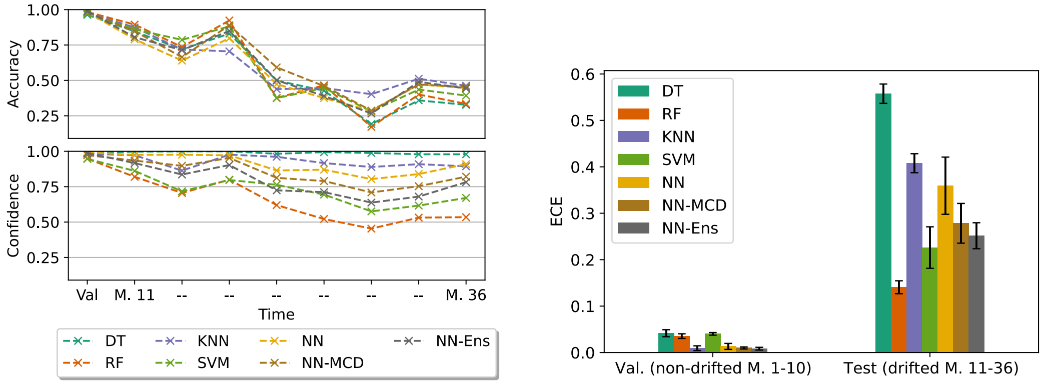
K-Nearest Neighbors (KNN) confidence estimates are calculated similar to DTs. The probability of a class is computed as the fraction of training samples of the same class in the set of nearest neighbors, weighted by their distance.

Random Forest (RF) confidence estimates are computed as the mean predicted class probabilities of the trees in the forest. The individual tree confidences are computed as described above (DT). This method of confidence estimation for RFs has been shown to be superior to more complicated extensions [22].

Neural Network For neural networks, we assess multiple recently proposed uncertainty estimation methods: **Max. Softmax Probability (NN)** [23]; **Deep Ensembles (NN-Ens)** [24] with $M = 10$ ensemble members. Randomness is introduced by reshuffling of the training set as well as different random initialization for each NN in the ensemble; **Monte-Carlo Dropout (NN-MCD)** [25] with $M = 20$ forward passes for each sample. Dropout rate p is set to 0.2.

2.2 Dataset

For our experiments, we use the Gas Sensor Array Drift dataset [9] that was recorded at the University of California San Diego (UCSD). The dataset was recorded over 36 months at an industrial test rig. Due to the long recording time, the dataset contains both sensor drift due to aging sensors and concept drift due to external influences which resembles the expected environment conditions of a real-world ML application in manufacturing, *cf.* Figure 1 (b). The dataset represents a classification task, in which the target variable is the type of gas (one of six) that is currently present in the apparatus. The experiments are perceived using 16 sensors and each row of the dataset contains 128 extracted statistical features (8 per sensor) of the corresponding experiment run with a total of 13,910 runs. The dataset is split into 10 consecutive batches, each capturing a varying amount of months.



(a) **Top:** Prediction accuracies over time. **Bottom:** Model confidences over time. *Val* indicates the validation set containing non-drifted data.

(b) ECEs of the assessed models for the non-drifted validation set (months 1-10) as well as averaged over the drifted test set (months 11-36). Error bars show the standard deviation of the experiment runs.

Figure 2: Results of the experiments on the UCSD Gas Sensor Array [9] dataset regarding model accuracy, confidence and calibration under drift. All experiments are repeated 10 times with different random seeds and the results consequently averaged.

2.3 Experiment Setting and Metrics

We train all the models on a random 50% split of the first 10 months of the available data and use the remaining 50% as the validation set for performance evaluation on non-drifted data. To be able to assess the robustness to drifts, we test on the remaining 26 months. All available features are used for training and the experiments are repeated 10 times with varying random seeds.

For evaluation, we employ two metrics capturing different aspects of interest:
Accuracy $\uparrow$ is used to assess the performance of the model on the non-drifted test set as well as the performance degradation under drift. The accuracy measures the percentage of correct classifications.
Expected Calibration Error (ECE) $\downarrow$ [26] is used to evaluate the calibration of the confidences produced by the model. The ECE is closely related to calibration curves and corresponds to the average gap between model confidence and achieved accuracy. While the ECE has several shortcomings [8], we choose it over other calibration metrics for its simplicity and interpretability to strengthen the practical relevance.

2.4 Results

As visualized in the upper part of Figure 2 (a), the classification performance of all tested algorithms strongly degrades with an increasing time difference to the non-drifted validation set. This indicates, that none of the tested algorithms is robust against drifts in the environment. Thus, online monitoring and eventual model updates would be required to guarantee a reliable and safe application in real manufacturing environments. In parallel, the reduction in accuracy is reflected by the lowering in confidence of a subset of the algorithms, most pronounced with RF. These confidences may be used to identify drift in this scenario using frameworks such as [27]. The calibration of the model confidences is further analyzed in Figure 2 (b). Notably, all tested algorithms show well-calibrated confidences on the validation set, reflected by the low ECE, while the calibration strongly degrades for the drifted data. In addition, the error bars in Figure 2 (b) show, that the standard deviation with respect to the ECEs of the 10 experiment runs on the drifted data increased for all algorithms when compared to the standard deviation on the validation set. This further highlights, that the calibration of a model on drifted data, can usually not be inferred from the calibration on in-distribution data as it highly depends on the type and magnitude of the drift. Lastly, it can be observed that the calibration of the RF degrades least for the drifted data, followed by SVM and NN-Ens, supporting the visualization in Figure 2 (a). Depending on the application requirements, the confidence of the RF may be used as a measure of the performance that can be expected of the algorithm as well as an indicator for drift. The observed comparably high calibration robustness of ensemble methods in the presence of drift aligns with previous work on deep ensembles [8]. A possible reason may be, that each of the models in the ensemble slightly overfits on different aspects of the training data, which in combination yields lower confidences on the drifted data, as the predictions will deviate more strongly than on in-distribution data.

3 Conclusion and Outlook

In this work, we highlighted the general relevance, implications and possible sources of drifts affecting the continuous reliability of ML applications in the manufacturing domain. Using an industrial dataset, we exemplarily show, that none of the most commonly used ML algorithms in manufacturing are robust against drifts in the data distribution inflicted by the environment or the sensors used for perception thereof. A consequent analysis regarding the confidence calibration of the algorithms showed, that in the majority of cases, the calibration strongly degrades with the drift, rendering the confidences unexpressive. Positively, the confidence calibration of ensemble algorithms such as random forests degrades less strongly and may be used to estimate the current performance and identify drifts. In a continual learning setup, the confidence could thus be used as a trigger signal for data collection and retraining of the respective model.

There are multiple opportunities for future work on drift detection and adaption through means of continual learning or domain adaptation specific to ML use cases in manufacturing such as condition monitoring, predictive maintenance or quality prediction to enable further adaption of ML applications in this domain. Especially the practical implementation of such systems on the shop floor level is still an open research issue.

Acknowledgments and Disclosure of Funding

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No. 958357 (InterQ), and it is an initiative of the Factories-of-the-Future (FoF) Public Private Partnership.

References

[1] 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. volume 2, 2021.

[2] 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.

[3] Yeounoh Chung, Peter J Haas, Eli Upfal, and Tim Kraska. Unknown examples & machine learning model generalization. *arXiv preprint arXiv:1808.08294*, 2018.

[4] Nicolas Jourdan, Eike Rehder, and Uwe Franke. Identification of uncertainty in artificial neural networks. In *Proceedings of the 13th Uni-DAS eV Workshop Fahrerassistenz und automatisiertes Fahren*, volume 2, 2020.

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

[6] Xinyang Wu, Mohamed El-Shamouty, and Philipp Wagner. White Paper: Dependable AI. Using AI in Safety-Critical Industrial Applications. Technical report, Fraunhofer Institute For Manufacturing Engineering and Automation (IPA), 2021.

[7] Beatriz Bretones Cassoli, Nicolas Jourdan, Phu H Nguyen, Sagar Sen, Enrique Garcia-Ceja, and Joachim Metternich. Frameworks for data-driven quality management in cyber-physical systems for manufacturing: A systematic review. *CIRP Conference on Intelligent Computation in Manufacturing (ICME)*, 2021.

[8] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua V Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *arXiv preprint arXiv:1906.02530*, 2019.

[9] Alexander Vergara, Shankar Vembu, Tuba Ayhan, Margaret A Ryan, Margie L Homer, and Ramón Huerta. Chemical gas sensor drift compensation using classifier ensembles. *Sensors and Actuators B: Chemical*, 166:320–329, 2012.

[10] Indrė Žliobaitė. Adaptive training set formation. 2010.

[11] Indrė Žliobaitė, Mykola Pechenizkiy, and Joao Gama. An overview of concept drift applications. *Big data analysis: new algorithms for a new society*, pages 91–114, 2016.

[12] Shailesh Tripathi, David Muhr, Manuel Brunner, Herbert Jodlbauer, Matthias Dehmer, and Frank Emmert-Streib. Ensuring the robustness and reliability of data-driven knowledge discovery models in production and manufacturing. *Frontiers in Artificial Intelligence*, 4:22, 2021.

[13] Guangfan Gao, Heping Wu, Liangliang Sun, and Lixin He. Comprehensive quality evaluation system for manufacturing enterprises of large piston compressors. *Procedia engineering*, 174:566–570, 2017.

[14] Fredy Sanz, Juan Ramírez, and Rosa Correa. Fuzzy inference systems applied to the analysis of vibrations in electrical machines. *Fuzzy Inference Syst. Theory Appl*, 2012.

[15] Nicolas Jourdan, Tobias Biegel, Volker Knauthe, Max von Buelow, Stefan Guthe, and Joachim Metternich. A computer vision system for saw blade condition monitoring. *CIRP Conference on Manufacturing Systems (CMS)*, 2021.

[16] Tom Diethe, Tom Borchert, Eno Thereska, Borja Balle, and Neil Lawrence. Continual learning in practice. *arXiv preprint arXiv:1903.05202*, 2019.

[17] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

[18] Simon Fahle, Christopher Prinz, and Bernd Kuhlenkötter. Systematic review on machine learning (ml) methods for manufacturing processes–identifying artificial intelligence (ai) methods for field application. *Procedia CIRP*, 93:413–418, 2020.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[20] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[21] John Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[22] Henrik Bostrom. Estimating class probabilities in random forests. In *Sixth International Conference on Machine Learning and Applications (ICMLA 2007)*, pages 211–216, 2007.

[23] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[24] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *arXiv preprint arXiv:1612.01474*, 2016.

[25] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[26] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[27] Lucas Baier, Tim Schlör, Jakob Schöffer, and Niklas Kühl. Detecting concept drift with neural network model uncertainty. *arXiv preprint arXiv:2107.01873*, 2021.

A Appendix

A.1 Hyperparameters used in the experiments

All input features are standardized using statistics calculated on the training set. Implementation is done using [19] and [20]. Only non-default values are reported here:
Neural Network The results are reported for a fully connected neural network with 3 hidden layers consisting of $32, 16$ and 8 units respectively. The hidden layers use Rectified Linear Unit (ReLU) activations, while the class scores are calculated using the softmax function in the final layer. Cross-entropy loss is used in training with the ADAM optimizer for 50 epochs at a learning rate of 0.001. **Deep Ensembles (NN-Ens)** utilize the same architecture for the ensemble models. $M = 10$ models are used in the ensemble. No dropout layers are used in the default case. For **Monte-Carlo Dropout (NN-MCD)**, dropout layers are added after each of the first two hidden layers with $p = 0.2$. $M = 20$ forward passes are performed for each sample. **Decision Tree (DT)** The decision tree is fitted with a maximum depth of 6. **Random Forest (RF)** The random forest uses 100 single estimators in the ensemble.