On Automatic Machine Learning for Industrial Condition Monitoring

Dissertation
zur Erlangung des Grades
des Doktors der Ingenieurwissenschaften
der Naturwissenschaftlich-Technischen Fakultät
der Universität des Saarlandes

von

Tizian Schneider

Saarbrücken
2024

Tag des Kolloquiums:	19.07.2024
Dekan:	Prof. Dr. Michael Vielhaber
Berichterstatter:	Prof. Dr. Andreas Schütze
	Prof. Dr. Robert Schmitt
Vorsitz:	Prof. Dr. Dirk Bähre
Akad. Mitarbeiter:	Dr. Amine Othame

Abstract

This thesis studies the question of how to utilize automated machine learning in industrial condition monitoring. The typical issues encountered are addressed, and a modular machine-learning approach is developed to solve them. Namely, it is easily applied with little machine learning knowledge due to full automation. It applies to various condition monitoring scenarios due to mutually complementing algorithms and toolbox-like structures. Its results are physically interpretable, creating an extra layer of trust and allowing deeper process understanding, while the prediction quality is on par with neural networks in robustness tests. Furthermore, the approach facilitates deployment on low-cost, high-efficiency edge hardware close to the sensors. That, in turn, reduces energy costs and required communication bandwidth compared to cloud computing. Additionally, the approach includes novelty detection and concepts that utilize it for outlier detection, monitoring of supervised learning, and detection of previously unknown faults. All named capabilities have been extensively and successfully compared to other approaches in different exemplary application scenarios. This success started the Data Engineering and Smart Sensors group at ZeMA and the Lab for Measurement Technology that further researched and extended the approach, e.g., by traceable uncertainty estimation following the Guide to the Expression of Uncertainty in Measurement.

Kurzfassung

Diese Thesis beschäftigt sich mit der Frage, wie automatisiertes maschinelles Lernen für industrielle Zustandsüberwachung eingesetzt werden kann. Ausgehend von dabei typischerweise auftretenden Problemen wird ein automatisiertes Konzept zu deren Lösung entwickelt. Es ist durch die Automatisierung ohne tiefes Verständnis maschinellen Lernens einsetzbar. Weiterhin deckt es mit sich gegenseitig ergänzenden Algorithmen und einer offenen Baukastenstruktur ein breites Anwendungsspektrum ab. Die gelernten Modelle sind physikalisch interpretierbar, was zu ihrer Vertrauenswürdigkeit beiträgt und den Aufbau zusätzlichen Prozessverständnisses ermöglicht. Gleichzeitig ist ihre Robustheit der von neuronalen Netzen gewachsen. Das Konzept kann auf kostengünstiger Rechenhardware direkt am Sensor umgesetzt werden, was im Vergleich zu Cloud-Computing notwendige Bandbreite und Energiebedarf reduziert. Darüber hinausgehend werden Konzepte zur Anomalieerkennung entwickelt, die Ausreißerdetektion, Überprüfung des überwachten Lernens oder Erkennung bisher unbekannter Schäden ermöglichen. Alle genannten Fähigkeiten wurden in mehreren Anwendungen mit denen anderer Konzepte verglichen. Die erzielten Erfolge führten zur Entstehung der Gruppe für Data Engineering and Smart Sensors am ZeMA und am Lehrstuhl für Messtechnik, in der diese Forschung weitergeführt und ausgebaut wurde. Zum Beispiel wurde die Berechnung der Messunsicherheit nach Guide to the Expression of Uncertainty in Measurement erweitert.

Table of Contents

1 Introduction

The ongoing process of the fourth Industrial Revolution (Industry 4.0 [1]), among other things, aims for better knowledge and control over production facilities [2]. Sensors, measurement science, and smart evaluation are the keys to achieving this goal. Multiple puublications have recognized and acknowledged this [3, 4]. With the industry's increasing focus on industrial services and value to the customer, future sensors have to emphasize the value of the measured data to the customer [5]. The goal is not to measure data but to gain additional process insight by extending the measurement chain by interpreting the measured raw data using machine learning [6]. A simple example of such an interpretation is the extraction of fault symptoms from a vibration sensor signal that allows deducing the remaining useful lifetime of a machine that could be used to schedule condition-based maintenance and reduce repair costs and downtime [7, 8]. This deduction process and identifying fault symptoms can at least be partially automated by machine learning [9] and was demonstrated among other works in the project iCM-Hydraulic [10]. iCM-Hydraulic followed the new measurement paradigm of condition monitoring (CM) using data-based modeling instead of physical sensor modeling. This paradigm has been identified as one of the major trends in measurement science [11].

The main research question for this thesis is how to apply this paradigm in a wide range of condition monitoring scenarios and how the various encountered issues can be solved simultaneously. More specifically, the questions are which algorithms and hyperparameters will be chosen for damage detection, how these algorithms are applied in real scenarios, and how novelty detection will support that. Note that all those questions have to be answered within the limitations of industrial conditions monitoring like vast amounts of recorded data, limited bandwidth to the cloud, limited computing resources on the edge, limited availability of machine learning experts, low trust in black-box models, very diverse sensors signals and high requirements for robustness.

2 Thesis Structure and Publications

Modernizing industrial smart sensors that offer process insights beyond their direct measuring capability by interpreting their signals poses a considerable potential for optimization, monitoring, control, and quality control. Section 3 outlines the vision for such a sensor and derives the resulting need for an automatic machine learning toolkit. Section 4 reviews typical applications expected for smart sensors, the respective difficulties for machine learning, and the current state of the art. Note that most of the discussed issues can be solved individually by off-the-shelf approaches; however, they lack performance when faced with regularly encountered combinations of problems.

Afterward, Section 0 constitutes the core of this thesis and introduces an automated machine learning approach of algorithms that were specifically chosen to cater to the needs of smart sensors in Papers 1-3 with goals described in Paper A. Additionally, Papers 1-3 show the approach's broad applicability, its application to novelty detection and comparisons to other algorithms.

Subsequently, Section 6 presents further research led by the author of this thesis that analyses and extends the suggested approach in Papers B-E and additional research projects based on Papers 1-3. Namely, this section shows machine learning limitations regarding domain shifts in Paper B. It complements the introduced toolbox's predictions with a metrological analysis of effects like measurement uncertainty and inaccurate sensor synchronization in sensor networks in Papers C and D, respectively. Paper E then extends the field of applications of the algorithm toolbox to structural health monitoring before the integration into no-code graphical user interfaces is shown. Additional algorithmic capabilities complement these extensions.

Finally, Section 7 reviews the achieved results and discusses the yet unsolved transferability problem and possible approaches to tackle this in future work.

2.1 Appended Papers and Author's Contribution as part of the Thesis

Three of the author's papers, formally included in the thesis, are listed in this section. They represent the main answer on achieving the goals and answering the research questions formulated in Section 1. Papers 1 and 2 answer those questions by introducing an automated machine learning toolbox for condition monitoring and its principles. Paper 3 highlights its extension to Novelty Detection. These papers are supplemented by other papers and further research by the author, described in the next section. Those are not part of the thesis but highlight the significance of the automated machine learning toolbox and provide a deeper understanding of the toolbox and its applications.

Paper 1 T. Schneider, N. Helwig and A. Schütze, "Automatic Feature Extraction and Selection for Classification of Cyclical Time Series Data," *tm – Technisches Messen* (2017)

I designed the automated machine learning toolbox, designed and performed the shown evaluations, and wrote the main part of the manuscript.

Paper 2 T. Schneider, N. Helwig, and A. Schütze, "Industrial Condition Monitoring With Smart Sensors Using Automated Feature Extraction and Selection," *IOP Measurement Science and Technology* (2018)

I extended the automated machine learning toolbox, performed most of the shown evaluations and model analysis, and wrote the main part of the manuscript.

Paper 3 T. Schneider, S. Klein, and A. Schütze "Machine Learning in Industrial Measurement Technology for Detection of Known and Unknown Faults of Equipment and Sensors," *tm – Technisches Messen* (2019)

I designed the three approaches to the different novelty detection applications shown, designed and performed the shown evaluations, and wrote the main part of the manuscript.

2.2 Other Appended Papers Based on the Author's Work

This section lists papers that are within the direct scope of this work but are not an official contribution. These papers highlight the author's continued scientific work after the research shown in the previous section as head of Data Engineering and Smart Sensors at the lab for measurement technology at Saarland University and ZeMA, which is highlighted by the position as the last author in most of these papers. Those papers and the corresponding research projects based on the automated machine learning toolbox will be discussed in detail in Section 6.

Paper A A. Schütze, N. Helwig and T. Schneider, "Sensors 4.0 – Smart Sensors and Measurement Technology Enable Industry 4.0," *Journal of Sensors and Sensor Systems* (2018)

Paper B P. Goodarzi, A. Schütze, and Tizian Schneider, "Comparison of Different ML Methods Concerning Prediction Quality, Domain Adaptation, and Robustness," *tm – Technisches Messen* (2022)

Paper C T. Dorst, T. Schneider, S. Eichstädt and A. Schütze, "Uncertainty-Aware Automated Machine Learning Toolbox," *tm – Technisches Messen* (2023)

Paper D T. Dorst, Y. Robin, S. Eichstädt, A. Schütze, and T. Schneider, "Influence of Synchronization Within a Sensor Network on Machine Learning Results," *Journal of Sensors and Sensor Systems* (2021)

Paper E C. Schnur, P. Goodarzi, Y. Lugovtsova, J. Bulling, J. Prager, K. Tschöke, J. Moll, A. Schütze and T. Schneider, "Towards Interpretable Machine Learning for Automated Damage Detection Based on Ultrasonic Guided Waves," *Sensors* (2022)

A complete bibliography of my research up to date is given in Appendix A. It includes 17 peer-reviewed journal articles and 36 conference papers.

3 Vision Sensor 4.0

Programmable sensor interfaces can already be purchased from suppliers like National Instruments [12]. However, for broad applicability, they need to be modularized and complemented by an equally modularized algorithm toolkit that can process data from various sensor types. In addition to condition monitoring of production equipment [13] or remaining useful lifetime services integrated into products [14], such a modular approach also opens application scenarios like in-process quality control in production [15] and assembly [16] as well as soft-sensing scenarios [17].

Given the high frequency of fault symptoms like ultrasonic vibrations [18] or current fluctuation [19], the software modules must allow training with big data [20] from various sources at high sample rates as well as support for hardware accelerated inference on the edge. The on-edge inference is needed to avoid overloading higher system components or communication networks with the full data streams.

Furthermore, automated machine learning must provide explainable models and the necessary trust for business decisions made on those models [21]. It also needs to support novelty detection to overcome the issue of previously unseen machine faults [22].

Condensed the Sensor 4.0 with machine learning capabilities needs software toolkits that offer:

- A wide range of application scenarios for classification and novelty detection
- Model explainability
- The capability to reduce a massive amount of data to highly relevant features on the edge
- Easy system integration on edge-hardware
- Training support for BIG Data

Such a toolkit was developed in Papers 1-3.

Besides the technical aspects of Sensor 4.0, the development process of such a sensor also has to consider economic elements. An exemplary estimate was given by Sosale and Gebhardt [23] for condition monitoring of pump aggregates that compares the economic prospects of run to failure, manual condition monitoring with regular checkups, and automatic condition monitoring with smart sensors. Depending on the industry and local wages, the maximum justifiable cost for automatic condition monitoring over manual condition monitoring ranges from 30 € (oil and gas industry in Asia) to 310 € (oil and gas industry in the North Sea) per sensing point. In every

calculated scenario, manual condition monitoring was preferred over run-to-failure strategies [23].

Some application partners in research projects like MoSeS-Pro, KI-Predict, and KI-MUSIK4.0 target lower prices, especially for smart sensor integration into products that offer self-monitoring capabilities. Note that by most companies, condition monitoring integration into products is usually seen as more price-sensitive than condition monitoring of production facilities.

In the context of Sensor 4.0, these costs include, in addition to the costs of the sensor itself:

- Installation, power supply, and communication
- ML model training
- ML model inference hardware

Therefore, costs have to be reduced in all these fields to achieve maximum scalability, which is one of the most critical factors for introducing AI in big companies in Germany [24].

Sensor costs: As will be described in more detail in Paper A, over-instrumentation during training data acquisition and subsequent virtual sensor removal or signal augmentation can be used to determine the minimal feasible amount of sensors [25]. The minimum required sampling rate [26] and minimal signal quality needed, i.e., the possibility of using low-cost sensors [25], can be determined similarly. This data reduction enables the deployment of a suitably scaled-down sensor network or single sensor. An example is shown in Paper 2.

Installation and wiring costs for energy supply and communication: Wiring costs can be reduced by using wireless sensors. Examples of such wireless sensors are the ABB Ability™ Smart Sensors or the Bosch CISS (connected industrial sensor solution) that transmit, among others, vibration data via Bluetooth™ Low Energy to smartphones or IoT Gateways. Combined with a sleep mode that records data only for a couple of seconds every hour, they are available with a battery life of up to 15 years [27]. Alternatively, such sensors could be powered by energy harvesting [28]. For ABB Ability™ Smart Sensors, the installation cost is minimized by utilizing magnetic mounts or special glue to prevent drill work. An alternative to installing dedicated condition monitoring sensors is retrofitting a data evaluation system on top of existing sensors, as shown in project iCM-Hydraulic [10]. Note that limited data bandwidth on low-power wireless communication protocols and the energy cost per

transmitted byte again motivate on-edge data reduction by feature extraction, as suggested in Papers 1 and 2.

ML model building: Utilizing machine learning for condition monitoring can eliminate the need for extensive physical modeling [14]. However, this advantage must not be traded for the need for extensive algorithm modeling development for particular use cases [8]. Therefore, a versatile, automatic ML approach that can be automatically adapted to different use cases is needed. Research and development of such an approach are crucial contributions of this thesis and are shown in Papers 1-3.

ML model inference: To fully use the automatic ML approach motivated above, model inference (or at least feature extraction and selection) must be integrated into the sensor system [29]. This can be done using low-cost microcontrollers or ASIC-based AI accelerators, as researched in projects KI-Predict [30] and KI-MUSIK4.0 [31]. Such a system reduces the data load necessary to capture information from high-frequency fault symptoms and energy consumption of the wireless data connection and, therefore, extends battery life with a suitably optimized accelerator. On the downside, the limited resources available for edge computing impose additional constraints on the ML modeling algorithms considered in Paper 1.

Training data acquisition: The gold standard for training data acquisition would be an over-instrumented pilot project [32]. Experiments would cover all possible operation modes and cross-influences on the sensor data and all their combinations to learn the minimal sensor setup needed to achieve the required robustness [25]. However, only data from in-line experiments in pilot plants are usually available, i.e., a pilot plant is retrofitted with sensors, and data is recorded during regular operation, leading to highly imbalanced datasets heavily favoring failure-free operation [29]. This data situation naturally leads to Novelty Detection, as discussed in Paper 3. Combined with embedding into a framework that starts with novelty detection on the regular operation, which can later be used for much more sensitive detection and quantification by supervised machine learning, maximum user profit is achieved. Since such a framework would require extensive research that could be carried out in another thesis, this thesis only focuses on supervised ML (Papers 1 and 2) and the algorithms suggested for novelty detection (Paper 3). Since data from regular operations cannot be used to determine the features needed for fault detection, all raw data must be centralized during this process. On the positive side, this removes the necessity for online learning and inference for novelty detection, allowing for a broader spectrum of algorithms.

Note that there are multiple alternatives to the approach described above. For single components, careful design of experiment (DoE) on a testbed that covers all expected

cross-influences and a limited number of experiments based on Latin Hypercube Sampling is usually the better alternative to the data acquisition in pilot facilities. This is because of better control that allows for highly reduced experiment times. However, this approach is typically too costly for condition monitoring beyond the component level, and correctly reproducing typical operation environments on a testbed is likely to be complicated. Other alternatives are online learning approaches like one-shot learning [33]. Online learning is especially useful if a Sensor 4.0 is intended to replace an expensive reference system that can provide target data during training, i.e., parallel operation of training data acquisition and reference system. On the negative side, online learning further restricts the choice of algorithms, is usually only feasible for minimal models, requires an online reference as a target, and eliminates the insight a process expert could gain from analyzing stored training data [29].

4 Background, challenges, and current research

Note that most of the challenges discussed in this section can be solved individually by off-the-shelf approaches that will be discussed as part of the current research. However, those approaches usually lack performance when faced with regularly encountered combinations of issues. Papers 1-3 specifically focus on solving combinations of multiple challenges at once.

4.1 Applications and Data Properties

Since a smart sensor has to adapt to a large variety of different application scenarios with very different requirements for data analyses, this section introduces examples of machine learning problems. These are typically encountered in condition monitoring and quality control and are used to derive the necessary versatility and additional properties of the analysis approach. Note that this list of examples is not intended to be complete and only aims to show the most often encountered domains from which information needs to be extracted.

In the first example, helpful information is visible in small sections of the signal, whereas large portions of the sensor signal are irrelevant for condition monitoring, i.e., the information is located in the time domain. For example, Figure 1 shows the movement speed of an electromechanical cylinder working against a constant pulling force just after replacement and just before failure [34]. As shown, the speed profiles differ significantly during and shortly after the acceleration phase on the return stroke due to a delay of the worn-out cylinder. In this phase, the position-controlled cylinder is accelerated against the pulling load. As wear increases toward failure, the efficiency decreases, and the drive fails to achieve the targeted acceleration. This causes the cylinder to drive faster for longer to catch up to the controlled target position after acceleration. This deviation, called contouring error, is the stopping criterion during fault detection. Due to its load dependency, it is only visible during this short high-load acceleration phase, and a fully automatic machine learning algorithm cannot utilize the knowledge about the contouring error. It needs to catch those deviations independently of the rest of the signal. An example of such an algorithm is the combination of Adaptive Linear Approximation (ALA), which separates the signal into linear segments, and a feature selection algorithm that selects signal slope during

acceleration and signal mean after acceleration as relevant features for the remaining useful lifetime [35].

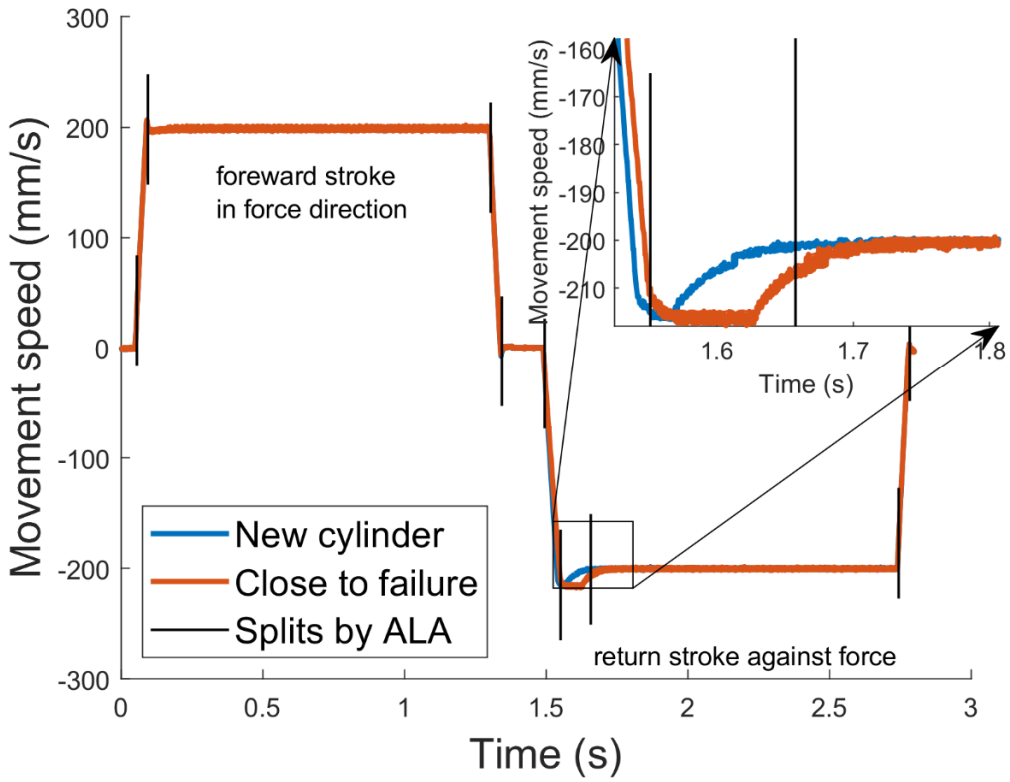


Figure 1: Velocity profile of electromechanical cylinder. At 1.5 seconds, the cylinder accelerates against a high load. With increasing wear, reaching the set speed and lag compensation to the controlled position is delayed. The wear has negligible effects throughout other parts of the cycle, as seen from the data of both lines overlapping everywhere except for the zoomed-out part. Note that this only holds up for this specific sensor, while other sensors (e.g., vibration sensors) can detect wear symptoms during different segments [26].

In some cases, the information relevant to condition monitoring is not located in one specific location but is apparent in the same way in multiple parts of the signal. One example is shown in Figure 2, which shows the pressure signal of a hydraulic machine that was set to hold six different pressure levels for ten seconds each [36]. The pressure control can achieve the targeted pressure independent of accumulator pre-pressure. However, accumulator leakage increases the time the machine needs to switch to the next pressure level. Since pressure switches occur multiple times, a localized feature extraction like ALA would need numerous features to capture all comprised information. In contrast, a global feature extraction method like Principal Component Analysis (PCA) [37] can capture the information from all switching processes simultaneously, which is visible in the coefficients of the first principal component that show upswings on rising pressure edges and downswings on falling edges. The result is a score on the first PC that is a very high quality (yet not perfectly linear) feature for accumulator leakage detection (see Figure 2).

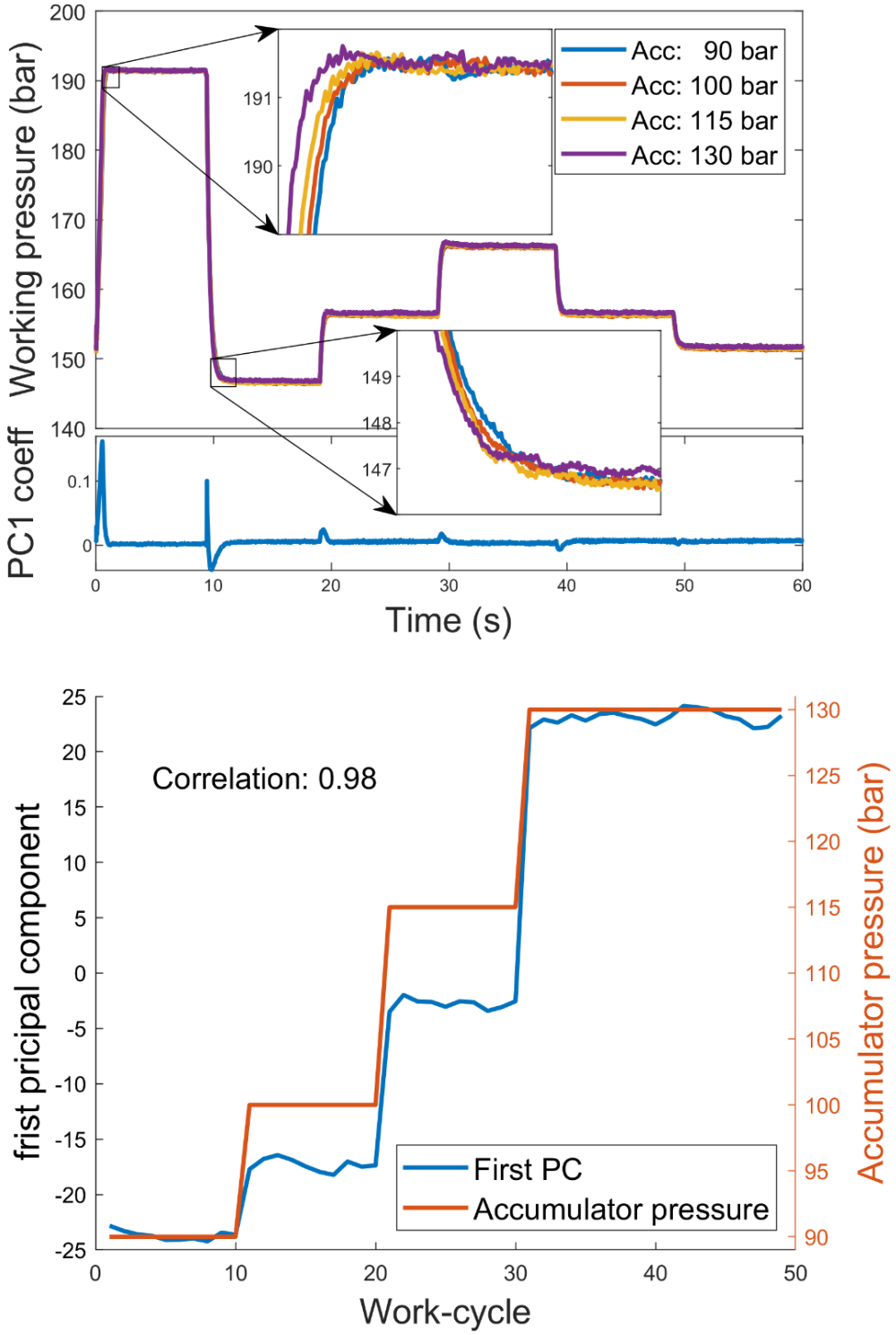


Figure 2: (top): Pressure cycle of a hydraulic machine with different accumulator pre-pressures. The higher the pre-pressure, the faster a new set pressure point can be reached. Given a dataset with multiple cycles from four different pressure steps and otherwise perfect working conditions, PCA can quickly identify this effect as the dominant factor in the dataset. Therefore, the coefficients of the first principal component reflect that by showing significant absolute values only around pressure switches. This also combines correlated information from multiple switches. (bottom): The resulting projection on the first principal component correlates strongly to the accumulator pressure and can be used to identify this fault.

For vibration signals typically recorded by a microphone or accelerometer, the fault information is usually not found in time but in the frequency domain. For example, Figure 3 shows the acceleration's absolute spectra of damaged and undamaged bearings [38]. Features useful for damage detection are oscillation energy in certain parts of the spectra, like damage frequencies of the inner ring, outer ring, or rolling elements. In this case, the expected defect frequency for the inner ring lies below 105 Hz [39] and nowhere close to 2.5 kHz, where Figure 3 shows the most oscillation energy for the damaged bearing. Therefore, in this case, the most significant differences cannot be retraced to the damage, highlighting the necessity for feature selection.

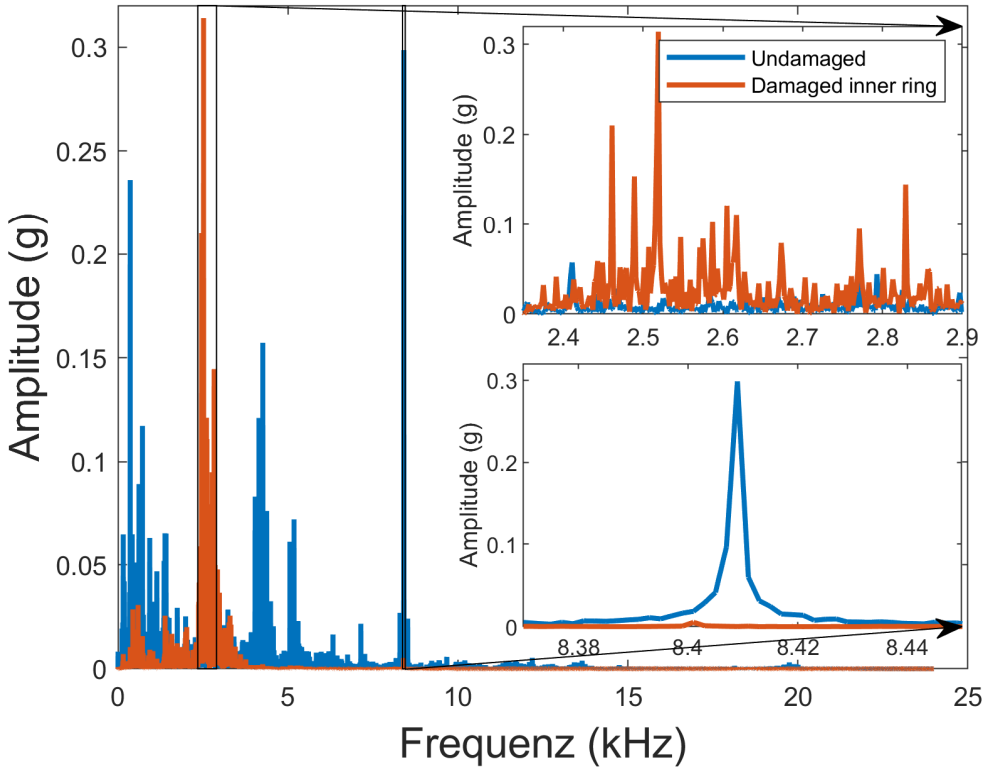


Figure 3: Two example spectra from the Case Western Reserve University bearing datasets with and without damage on the inner ring. Relevant signal differences are localized in the frequency domain around 2.5 kHz and 8.4 kHz (zoomed plots). Note that the authors do not offer any physical explanation of these differences [39].

Similarly to the time and frequency domain transformations shown above, many condition monitoring use cases have been shown to profit from the time-frequency transformation offered by Wavelet Transform that allows a multiresolutional view of the data [40].

The last example (see Figure 4) shows how relevant information can be derived from the statistical data distribution. It shows data from the same hydraulic machine used to create Figure 2. The signal was recorded using an accelerometer mounted on the main

pump [41]. As seen in Figure 4, the signal kurtosis on a pressure plateau can be a valuable feature in detecting severe accumulator pressure loss. The kurtosis can be interpreted similarly to the crest factor. A larger kurtosis means a more significant proportion of samples with high deviation from the mean, i.e., higher acceleration values. Note that this feature is noisier than the one depicted in Figure 2. Nonetheless, it is still preferable in some applications since it is robust against other cross-influences like valve sticking that would have a high impact on the PC score due to the valve switching delay it causes. Therefore, in this example, there is a strong argument for extracting features from the statistical properties of sensor signals.

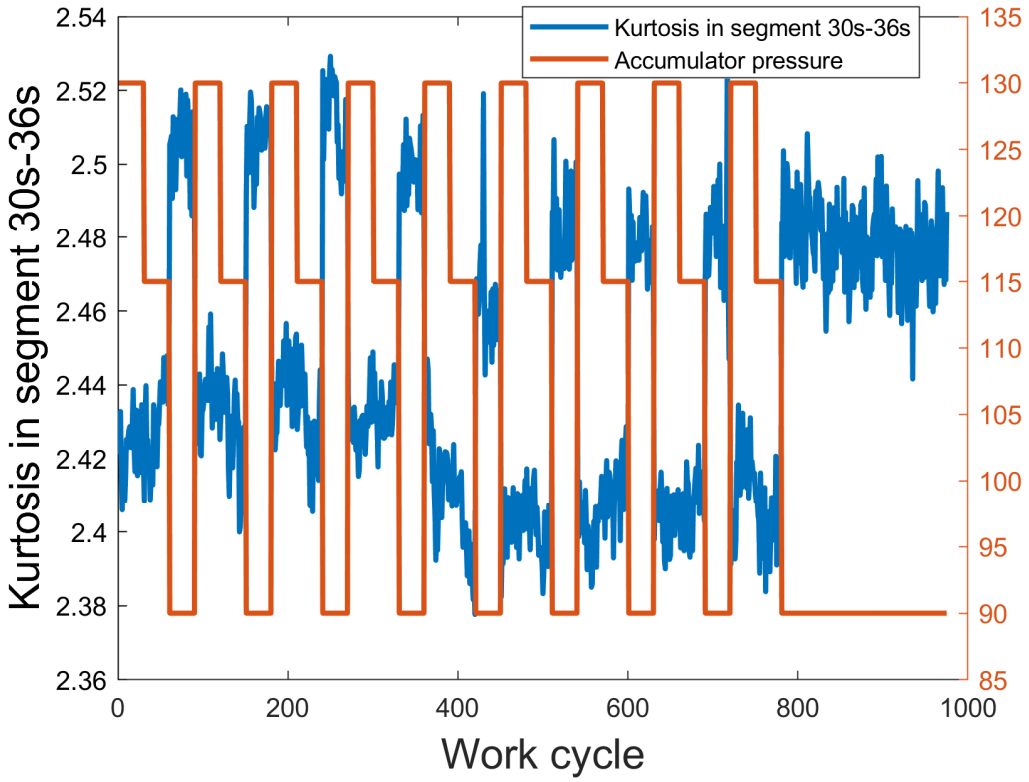


Figure 4: This figure shows the Kurtosis of vibration signal extracted from time segment 30s-36s and correlated accumulator pressure. The sensor is mounted on the main pump of a hydraulic system. The kurtosis can be used to identify severe loss of pressure.

In conclusion, the resulting feature extraction problems typically encountered in condition monitoring are too diverse to be solved by one feature extractor. Based on an extensive comparison and benchmark of multiple feature extraction algorithms [42], Papers 1-3 of this thesis suggest a carefully chosen combination of mutually complementing feature extraction algorithms. Papers 1-2 specifically tackle the problem of selecting a suitable extractor for supervised learning, while Paper 3 tackles the same problem for unsupervised learning.

4.2 Machine Learning Challenges

When applying ML to CM, one faces multiple data quality challenges. These arise from the data quality typically encountered in industrial applications and modeling challenges on this data. Since both these categories require extensive research to be covered exhaustively, this section only focuses on the biggest and most commonly encountered challenges in industrial CM. It derives requirements for the employed machine learning algorithms and their usage. In a typical CM scenario, multiple interfering challenges that mutually reinforce and/or mask each other are encountered.

4.2.1 Data Quality Challenges

In an open discussion preceding the MST-Congress 2021 [43], multiple industry and academic experts agreed that poor data quality is the biggest problem commonly encountered in industrial machine-learning scenarios, even in research projects on AI for predictive maintenance.

The most common problem with data quality is the **statistical dependency of training samples**. Although statistical learning theory is one of the foundations of machine learning and assumes data that are identically distributed and independently drawn [44], industrial data is usually recorded under constant conditions, making the data statistically dependent. However essential for ML, the topic is insufficiently treated in scientific literature. An example is the famous deep learning review by LeCun, Bengio, and Hinton published in Nature [45], which by March 2024 was cited more than 76,000 times [46]. This paper contains precisely two sentences about data. The first states, "We first collect a large data set of images, of houses, cars, people and pets, each labeled with its category" [45]. The second states, "In a typical deep-learning system, there may be […] hundreds of millions of labeled examples with which to train the machine" [45]. None of them state any requirements about data quality. Papers from the engineering community usually refer to data quality standards inspired by conventional measurement science. Since ML in condition monitoring is an extension of the measurement chain, those standards are essential, however insufficient for ML, as they neglect the statistical independence of data samples: E.g., the 15 IQ dimensions of data quality formulated by Hildebrandt et al. [47] does not include statistical independence as the dimensions "completeness" only takes into account missing data from individual sensors, and the dimension "sufficient for the current task" only names the number of training samples but not their statistical properties [47]. Statistical independence can only be achieved by running multiple experiments under multiple conditions during training data acquisition (e.g.,

environmental conditions, load conditions, repeated wear experiments with different test objects of the same type). Since such variations in experiments are either extremely expensive, can only be performed on testbeds, or are just plain impossible (e.g., fault simulation in retrofitting applications), the employed algorithms must be able to handle both redundant data sample from experiment repetition under similar circumstances and a meager number of independent samples. Therefore, the algorithms suggested in Papers 1-3 are chosen to be as robust and straightforward as possible to optimize their ability to generalize, which is confirmed in the studies performed in Paper B.

A second challenge often closely connected with statistically dependent training samples is the issue of **unrepresentative data**. One facet of unrepresentative data is the regularly encountered low number of (statistically independent) training samples. The principal problem lies in the law of large numbers. It states that the average of the results from many trials of the same experiment should be close to the expected value and tends to become closer to the expected value as more trials are performed [48]. In reverse, this law shows that extreme results (i.e., results far from the expected value) concerning statistical evaluations are more likely to appear in small sets of data samples. Concerning CM, that means that estimations of validation and test errors on small datasets might be far from the actual expectancy values, and the performance of a given algorithm cannot be measured reliably. Since this issue cannot be solved with statistical approaches, it requires global interpretability of the employed ML method to verify its significance with physical process knowledge. Therefore, all algorithms employed in Papers 1 and 2 offer global interpretability.

The same requirement of global interpretability can be derived from the correlation versus causality issue that arises from possible correlations between the target variable and other interfering variables that influence the sensor signals. If those interfering variables cannot be measured during training data acquisition, this problem is not detectable with purely statistical approaches and has to be solved by physical model interpretation. If the application allows a broad design of experiments, this problem can also be solved by the design of experiment that ensures statistical independence of controlled variables.

The last facet of unrepresentative data treated here is errors in datasets that occur in all machine learning applications. One example Meske et al. [49] show is the presence of source tags in ca. 20% of images of horses that were used to train image classifiers. The authors demonstrated the resulting issue by employing explainable AI methods that highlight the source tag in images classified as horse and the fact that classifiers cannot detect horses when source tags are removed. Also, they showed that classifiers would detect a horse if the same source tag is added to a random image, and the

explainable AI method will again highlight the source tag [49]. In condition monitoring, such errors usually arise from systematic measurement errors during training data acquisition. They can be solved by following the conventional rules of measurement science, such as using calibrated sensors. However, due to their frequent appearance, the ML method should be able to detect such errors. The above example shows that local model explainability is sufficient for this task. However, global explainability is preferred since it relieves the need to check each example individually.

The last data quality challenge addressed here is **the low accessibility and data availability** that complicates comparisons between different algorithms. This problem is manyfold and can have multiple origins prohibiting training data acquisition or access to training data. Examples include restrictive data protection policies by companies that do not allow open data, the need for expensive and proprietary interfaces to read sensor data from control units, or limitations of open interfaces like sample rate with OPC-UA. The biggest problem, however, is a missing coherent standard for data representation and file formats due to the vast variability in applications and respective data structures. This is in direct contrast to, e.g., image classification, where the excellent availability of data is one of the reasons for the success of deep learning methods. Even preinstalled tools like "Windows Photos" can handle 41 file formats [50] to depict or resave them coherently. This problem cannot be solved quickly, but it regularly has to be acknowledged. Although it does not impose direct requirements on the ML algorithms, it suggests that not all AI research on image or audio classification can be transferred to condition monitoring due to this missing key component of the success of deep neural networks. Therefore, Papers 1-3 do not suggest neural networks for condition monitoring, which is also justified by the studies in Paper B.

4.2.2 Modeling Challenges:

The first challenge a machine learning algorithm encounters during training for condition monitoring is the high dimensionality of training data. Most neural networks for image classification work on rescaled images from 224x224 to 331x331 pixels [51] (i.e., $1.5 - 3 * 10^5$ input dimensions for colored images). At the same time, in condition monitoring, the dimensionality of raw data is usually much higher due to the high sampling rates necessary for detecting high-frequency wear symptoms like higher harmonics in electrical currents [34]. For example, during one working cycle of the electromechanical cylinder mentioned in the previous section, nearly 10^8 measurements are recorded. Significant resampling is often possible after confirming

the absence or low relevance of high-frequency information [26]. This high dimensionality causes problems like overfitting [52, 53], i.e., interpretation of noise, and the curse of dimensionality [54, 55], showing diminishing measurement contrast in high dimensional space. Aside from the encountered “finding a needle in a haystack” problem, algorithms might suffer because the context of the target variable might be easier to detect than the target variable itself. As an example, consider the electromechanical cylinder use case. As shown in Paper 2, the remaining useful lifetime of a single cylinder is easy to predict with a model trained explicitly for this cylinder. At the same time, all three analyzed cylinders can be discriminated by an unsupervised, linear PCA on the feature level. This constellation allows any complex algorithm to learn a model tree that first detects the correct cylinder and then predicts its remaining useful lifetime with a cylinder-specific model. This model tree would appear very accurate in random cross-validation. However, it would not be transferable to new cylinders. It is essential to carefully choose training, validation, and test data to detect this. In this example, leave one cylinder out cross-validation would be appropriate. To prevent this issue, the model's capacity would have to be reduced, so model trees become impossible to learn for the algorithm, as is the case for the suggested toolbox from Papers 1-3.

The abovementioned issue is closely connected with the model challenge of strong cross-influences. The ML challenges in industrial datasets are usually relatively easy to solve due to typical wear symptoms like increased power consumption, decreased efficiency, or increased vibration intensity. At the same time, they can be complicated with cross-influences like temperature, load, speed, and others. In machine learning, those cross-influences are called domain shifts. For the datasets analyzed in this thesis, they are so significant that they are treated in Paper B as a separate issue. It should be mentioned that current research suggests simpler methods are less susceptible to domain shifts [56].

In total, there are three conclusions to be drawn from the challenges mentioned above:

1. Global interpretability is a massive advantage for algorithms used for CM
2. Robust methods should be preferred over complex and powerful methods
3. Training and validation methods need to handle the low statistical independence of data samples

These conclusions explain the use of strictly linear – and therefore inherently and globally explainable – classification with physically explainable features in Papers 1 and 2. At the same time, linear models are the simplest and, therefore, most robust

models available to solve the problems at hand (see Paper 2). They especially excel when tested for generalization in the presence of statistical dependencies (Paper B).

4.3 Data Science Approaches

Proceeding from the data, applications, and challenges shown in the previous section, three major scientific communities research machine learning from such or at least very similar data. Partitioned by their primary field of research, those are:

- Mechanical engineering community
- Time series data mining community
- Deep learning community

Having very different scientific backgrounds, each community developed its own solutions for the problems at hand that, in turn, offer different advantages and disadvantages for the use in condition monitoring. Those solutions will be reviewed and discussed in the following.

Mechanical engineering community: ML in the mechanical engineering community evolved from conventional (physical model-based) signal processing and typically utilizes conventional machine learning pipelines comprised of feature extraction, dimensionality reduction (feature selection and projections), and classification or regression [57, 58]. Note that there has been and still is a smooth transition of this community towards deep learning algorithms that are increasingly used for condition monitoring [57]. However, the main focus of this community is feature extraction. The most common transformations whose properties are used for feature extraction are Fast Fourie transformation (FFT) [40, 59, 60], wavelet transformation (WT) [60], and short-time Fourier transform (STFT) to transform signals from acoustic emission, vibration or force sensors [57]. Also, signal statistics like statistical moments, root mean squared, crest factor [61], and parameters of fitted autoregressive integrated moving average (ARIMA) models [62] are commonly used as features. Typically, algorithms are suggested for a specific target application and incorporate substantial physical domain knowledge of this specific application. A simple example is the damage frequencies of ball bearings [63]. The most common ML algorithms trained on these features are (shallow) neural networks, Support Vector Machines, and Decision Trees [58].

Concerning sensor 4.0, the advantages of this physically motivated approach of this community are:

- All algorithms directly target and are designed for condition monitoring.
- Domain knowledge leads to highly sensitive and robust features.
- Good physical interpretability of features.
- Features are usually computationally cheap.

The disadvantages are:

- Features are tailored for specific applications and require detailed system knowledge.
- Suggested algorithms are typically tested on single, unpublished datasets generated by the proposing authors themself and are insufficiently compared to other algorithms [64]. While this is forgivable concerning the difficulty of obtaining such datasets, especially concerning privacy issues arising from real-world machine data, this lack of commonly available datasets is considered one of the most pressing issues in CM model training [65].
- A study found that “Many models reported in the literature lack proper validation procedures [… that are …] now viewed as standard in other domains” [58]. This issue leads to algorithms with unknown generalization performance.

Time series data mining community: The development of the time series classification community dates back at least to 1993 [66]. However, a review of the first decade of research showed that most papers used only a single artificial dataset created by the proposing authors themselves for testing [64]. Motivated by the need for comparable benchmarks, Keogh and Folias created the UCR Time Series Classification Archive in 2002 [67]. They extended it in 2015 [68] and 2019 [69] to 129 time series datasets with identical data structures. With more than one thousand published papers using at least one of the datasets from the archive, it became the core of the time series data mining community [69].

The research on the UCR repository showed that a simple "one-nearest-neighbor with Dynamic Time Warping (DTW) distance is exceptionally difficult to beat" [70]. Since 2013, the research has focused on ensemble methods that transformed the time series into a new feature space, e.g., using shapelets transform [71]. This research culminated in the proposal of HIVE-COTE, an algorithm that ensembles 37 different classifiers over multiple different time series representations and combines those ensembles with a hierarchical structure and probabilistic voting system [72, 73]. Two independent and extensive benchmarks of different algorithms on the UCR archive showed HIVE-COTE to be the best-performing time series classification algorithm [74, 75].

Concerning sensor 4.0, the advantages of this time series classification approach and similar approaches from this community are:

- High versatility that has been tested on a large variety of datasets.
- Fully automatic training without any manual hyperparameter tuning or algorithm selection.
- The classification performance of HIVE-COTE is unbeaten.

The disadvantages are:

- High dependency of HIVE-COTE and similar suggested algorithms on nearest neighbors that results in unrealizable (concerning cost constraints) high memory requirements during inference on the edge (training data needs to be stored).
- HIVE-COTE utilizes shapelet transformation with an algorithmic complexity $O(N^2 l^4)$ for N samples of length l [76] which results in an infeasible computational training cost for most of the targeted applications.
- The application domain of datasets from the UCR archive is biased towards datasets that reflect the personal interests/hobbies of its creators, datasets that could be easily obtained or created, and datasets that do not have privacy issues [69]. These fields do not include condition monitoring.
- The UCR archive exclusively comprises classification datasets, whereas sensor 4.0 applications, such as remaining useful lifetime estimations, also require quantification.
- The UCR archive comprises only small datasets (compared to training datasets expected in condition monitoring applications) with a maximum training set size of 1000 samples. The complete archive file size (training and test of 129 datasets) is only 853 MB in uncompressed ASCII file format and 301 MB as compressed zip [67]. Larger datasets that comprise a "finding a needle in a haystack" problem are not included.

Deep Learning community: The deep learning community is motivated by the success of Alex Krizhevsky's deep convolutional network in the ImageNet Large Scale Visual Recognition Challenge 2012 [77] and aims to adapt those results to other ML domains, including condition monitoring. Deep neural networks are currently the most widespread algorithm class in machine learning. For condition monitoring, most papers use deep multilayer perceptrons, long short-term memory, convolutional neural networks, or deep reinforcement learning [57]. Those architectures provide a complete end-to-end learning pipeline that automatically learns discriminative features from the

training data. The community is divided over whether the many parameters tuned during end-to-end learning lead to a high susceptibility to overfitting, as claimed by [78], or not, as claimed by [75]. Overall, the deep learning community is highly optimistic about the superiority of deep architectures over conventional machine learning; however, for condition monitoring, most architectures and hyperparameter configurations are again tested only on single, unpublished datasets created by the proposing authors [75].

Concerning sensor 4.0, the advantages of deep neural network architectures are:

- Neural networks are highly versatile and even universal in theory [79].
- They are suitable for hardware acceleration, and multiple hardware accelerators are available [80].
- There are countless studies claiming superiority over conventional ML algorithms [57].
- A benchmark on the UCR archive showed deep resNet architectures to be close second to HIVE-COTE [75] (providing statistically insignificantly worse performance) while being computationally cheaper in training and inference since both can be accelerated by hardware.

The disadvantages are:

- Unknown impact of the significantly worse dimensionality to the number of training samples ratio common in machine learning. As most common image classification nets have an input dimensionality of 224 pixels x 224 pixels x 3 color channels (e.g., googleNet [81], vgg19 [82], resnet101 [83]), i.e., 150.528 dimensions and are trained on the ImageNet database with more than 14 million labeled images [84], there are almost ten training samples per input dimension. At the same time, even the relatively small (in data recorded per sample) hydraulic condition monitoring dataset [36] has an input dimension of 50,000 measurements per sample and only roughly 1,500 samples, leading to 0.03 samples per input dimension.
- As will be shown in Paper B, deep architectures, in comparison to conventional ML algorithms, are more susceptible to commonly expected domain shifts and resulting deviations are more complex to compensate.
- Architectures and hyperparameter configurations are very difficult to choose.
- Results of neural network training and prediction are still hard to interpret, and explainable AI usually focuses on local explainability of individual predictions.

In conclusion, none of these approaches satisfies the requirements to be easily applied to condition monitoring.

The approach suggested in Papers 1-3 aims to combine the idea of ensemble learners ensembling over multiple different data representations dominant in the time series classification community with physically motivated data representations common in the mechanical engineering community. Therefore, a subset of five physically interpretable and computationally cheap transformations for mutually complementing data representations in time-domain, in time-frequency-domain, in frequency-domain, by statistical properties, and by signal segmentation were chosen. However, as the mechanical engineering community typically does, only the best representation is selected for inference to reduce hardware requirements during inference on the edge. Exceeding the thesis, the approach was combined with hyperparameter optimization techniques that originated from the deep learning community to further increase performance [85].

4.4 Issues of High Dimensionality

Independent of the specific algorithm used, all approaches named in the previous section are based on similarities of data examples within a given group (classes) and differences between groups, i.e., contrast. For regression, it is based on similarities of samples in close vicinity of the predictor variable value and differences between very different predictor variable values, respectively. Those similarities and dissimilarities must be derived from the statistical properties and data distribution of the training data during the training process to be recognized in model inference. The different learning algorithms constitute different approaches to this analysis and the maximization of contrast; however, they all suffer to some extent from the same issues of computational cost, overfitting, and curse of dimensionality when applied to the very high dimensional data shown in Section 4.1.

4.4.1 Computational Cost

The applied algorithm's computational cost and memory requirements must be considered for inference on the edge during algorithm selection. For the high dimensional data expected in typical target applications, both computational complexity and actual computational cost are essential to ensure scalability and applicability. An example where this differentiation is crucial is the memory cost of k-Nearest-Neighbor (KNN) that scales linearly in both the number of features and the

number of training samples but is still prohibitively large for the described applications due to the limited memory on the targeted low-cost edge devices. The computational complexity in the number of measurements per training sample is substantial during training. In contrast, complexity in the number of training samples can often be neglected due to their low number. However, for massive datasets like Festo electro-mechanical cylinder lifetime estimation, the availability of Map-Reduce training algorithms is highly advantageous. During inference on the edge device, one concern is the memory used to store the prediction model with all parameters and intermediate calculation results. Therefore, continuous (streaming) data processing algorithms are better suited than those requiring buffering the complete input sample (cycle) before the processing can start. All algorithms suggested for the first steps of data processing and reduction in Papers 1-3 were specifically chosen to be as scalable as possible.

4.4.2 Overfitting

Overfitting occurs in models with more parameters than appropriate for the available data [52]. It denotes the generation of a model that fits the data too closely due to the unintentional extraction of residual noise as if that noise represented the underlying model structure with the result of poor generalization performance [53]. One way to reduce the probability of overfitting is hyperparameters many algorithms utilize to tune the bias-variance tradeoff between model simplicity (bias-error caused by underfitting) and performance on the training set (variance-error caused by overfitting) [86]. Another way is to control the ratio between the number of features and the number of training samples. For example, it should be at least one in ten according to the widespread "one in ten rule" of thumb [87]. Note that low or no information content of the data and a mismatch between assumptions made by the learning algorithm about data distribution and the actual data distribution significantly increases the risk of overfitting [42]. After training, a model should always be checked for overfitting by validation (see Paper B). Due to the typically very high number of features in applications discussed here, underfitting - the opposite of overfitting - is usually irrelevant.

4.4.3 Curse of Dimensionality

The curse of dimensionality describes multiple phenomena that only occur in high-dimensional spaces and, for machine learning, ultimately reduces the quantifiable contrast between similar and dissimilar data samples and, therefore, model performance [55]. Mathematically, the curse of dimensionality is caused by the

volume of the feature space growing exponentially with the number of features [54] requiring an exponentially increasing number of samples to maintain constant sample density [88]. For example, consider two variables requiring 100 data samples to be well sampled. If combined, the variables would need 100x100=10,000 samples to be sampled equally well as the individual variables. In a very simplified manner, it could be understood as minor differences caused by random noise that accumulates to a significant difference even for similar data samples when many noisy features are used for model building. This is comparable to measuring slight variations (i.e., differences between different samples) on top of a large offset (accumulated minor differences between similar samples) in classical measurement science.

In conclusion of those difficulties, the automated machine learning approach introduced in Papers 1-3 emphasizes extensive dimensionality reduction and scalability of the feature extraction that constitutes the first step of dimensionality reduction. In total, there are four steps of dimensionality reduction. Namely feature extraction, univariate feature selection (preselection), multivariate feature selection, and projection by Fisher's Linear Discriminant Analysis [89], whereas the risk of overfitting is the main reason for feature extraction being performed unsupervised. To counter the curse of dimensionality, the first step of feature selection is performed by treating each feature individually before the multivariate feature selection in the next step considers feature interactions. The last dimensionality reduction step is the projection on discriminant functions (see Paper 1).

4.5 Feature Extraction

The first step of dimensionality reduction typically is feature extraction. It refers to calculating sensor signal characteristics from raw data that remove correlations between individual measurements within the signal and simultaneously describe the signal as closely as possible with as few features as possible. While doing so, each algorithm should preserve as much signal information as possible. For machine learning, similarities and differences between data samples should be preserved, i.e., similar (dissimilar) data samples should lead to similar (dissimilar) features. This requirement is known as the Lower Bounding Lemma [90]. When this lemma is fulfilled, Feature Extraction improves ML performance by mitigating the effects of overfitting and the curse of dimensionality. Additionally, it reduces input data correlations and concentrates information on a few relevant features, allowing for effective further dimensionality reduction by feature selection.

However, since the best compression of a given dataset, i.e., the best feature representation, is indeterminable [91], no feature extraction algorithm can be proven to be optimal for a given dataset. As a result, an automated feature extraction algorithm can only suggest but not prove how the data should be represented. Human domain experts can create higher-performance features using their domain knowledge.

Additionally, each feature extraction algorithm has to decide the tradeoff between good approximation and the number of features used since signal characteristics should be extracted. At the same time, random noise and irrelevant variations should be neglected. This tradeoff, again, cannot be decided in general [92]. It is very similar to deciding the bias-variance-tradeoff in model building because it involves deciding which data variations are model-based and which are random noise.

A typical feature extraction algorithm is designed to find a data representation that fits the signal characteristics as closely as possible. How well a given algorithm can capture the contrast between descriptive and noise-representing features highly depends on the dataset [64]. The three most relevant approaches to feature extraction in literature are physically motivated features, features learned by end-to-end learning algorithms like neural networks, and reconstruction-based features. The advantages and disadvantages of the first two approaches have already been discussed in Section 4.3.

The reconstruction-based algorithms aim to reduce the approximation error between the original signal and its corresponding reconstruction from feature representation. The basic assumption of those algorithms is that capturing the most dominant characteristics will significantly decrease the approximation error between the original and the reconstruction from features while capturing incompressible noise will slightly reduce the approximation error. One example is the extraction of Fourier coefficients with the highest signal energy from vibration signals.

The advantages of reconstruction-based algorithms are:

- Their capability to extract those features that are optimal in the algorithm's respective data representation concerning reconstruction error.
- Their unsupervised nature reduces the likelihood of overfitting.

The disadvantages of reconstruction-based methods are:

- The main characteristics might be unrelated to the actual ML target and the algorithm's inability to test for relevance due to the algorithm's unsupervised nature.

- Most algorithms cannot decide the tradeoff between low approximation error that will always decrease with more features and a low number of extracted features. In the best-case scenario, the number of features to extract can be determined heuristically.

Consequentially, in Papers 1-3, it is suggested to use reconstruction-based feature extraction methods that utilize complementary and physically motivated representations for automated ML for smart sensors. To increase the applicability of smart sensors, it is suggested to relinquish the advantages of ensembling that representation and to decide for only the most promising in the respective application. This suggestion makes unsupervised feature extraction algorithm selection impossible, and a complete benchmark of all considered feature extraction algorithms is required to ensure optimal algorithm selection. The suggested algorithms and their respective heuristic to decide the number of features to extract are introduced and explained in Papers 1 and 3.

4.6 Feature Selection

Another method for dimensionality reduction is feature selection, which tries to select highly relevant features for the learning target and omit highly correlated or irrelevant features. In general, feature selection aims to choose the feature subset that results in minimal learning error. However, this de facto goal cannot be achieved due to a series of problems with feature selection. Since those problems can only be given a quick review in the scope of this thesis, the interested reader is referred to the excellent work of Guyon and Elisseeff [93] for more insight and striking examples.

The three most essential difficulties of feature selection and their consequences are:

- Features might only be relevant in combination with another feature or combinations of features that might or might not seem irrelevant without the context of the additional feature. That means the information content of a feature can only be evaluated in the context of other features [94]. All feature combinations must be tested for relevancy to determine the guaranteed best feature subset. This is usually impossible due to the exponential growth of the number of feature combinations and corresponding testing time with the number of features to choose from.
- Feature correlation does not imply feature redundancy [94]. Therefore, eliminating highly correlated features cannot reduce the exponential number of feature combinations.

- The features of the best combination of n features might not be part of the best combination of n+x features [94]. This so-called nesting problem prevents the optimal usage of feature ranking algorithms that, by design, cannot represent nesting problems.

Additionally, there is the general issue that even if there were an algorithm that could assign relevance scores to features despite the need to check an exponential amount of combinations, it would only allow to omit probably approximately irrelevant features. This is due to random variations that would prevent the relevance score from being exactly zero (approximately irrelevant) and the limited training data that allows only stochastic assessments (probably irrelevant) [93].

In conclusion, no (practical) feature selection algorithm can be optimal, and various algorithms with different advantages and disadvantages were developed for different problems. The algorithms can be categorized as filters, wrappers, and embedded methods [93].

Filter Methods: Filter methods are ranking methods that rank each feature by its individual information content concerning the learning target [93].

Advantages:

- Filter methods are swift and applicable to massive feature sets.
- The rankings of filter methods are easy to understand.
- Implementations of many methods used for feature ranking are available on various platforms and programming languages.

Disadvantages:

- Feature interactions are ignored.
- Feature redundancy and correlation are ignored, and filters tend to select highly correlated features.
- Filters try to select a generally good feature set instead of choosing the best subset for the specific learning algorithm used.
- The optimal number of features to choose from has to be determined outside the algorithm.

Wrapper Methods: Wrapper Methods are binary search algorithms that wrap around the subsequent learning algorithms to find the optimal subset for this specific learning algorithm. The search space is binary, and for n features, n-dimensional since each feature can either be part of the optimal set or not [94].

Advantages:

- The algorithm searches for the optimal feature subset for the learning algorithm.
- Both feature interaction and redundancy are handled concerning the learning algorithm.

Disadvantages:

- Wrapper methods are time-consuming compared to filters and embedded methods due to their need to train and validate the learning algorithm in every step and the exponential size of the search space.
- Many wrappers, and especially relatively fast algorithms, cannot resolve nesting problems.
- The search algorithms cannot guarantee a global minimum (except for exhaustive search). They might select unstable subsets, which is similar to overfitting, as such a subset of features would work well for the training data without generalizing to new data [94].

Embedded methods: Embedded methods are special variants of common learning algorithms that have been modified to base their decision process only on highly relevant features and to ignore less relevant ones [93].

Advantages:

- Embedded methods are typically faster than wrappers.
- They take into account feature interactions and redundancies.

Disadvantages:

- Embedded methods usually are very complicated, which makes their results hard to interpret.
- There is a wide variety of different embedded methods, and it is unclear how to select the best algorithm [42].
- For most embedded methods, the optimal number of features needs to be decided outside the algorithm.

Since no applicable algorithm can guarantee optimal performance [94], a combination of algorithms is chosen for feature extraction. Namely, RFESVM, RELIEFF, and Pearson correlation for feature ranking is suggested in Paper 1 and Paper 2. This suggestion is predominantly based on the author's Master thesis that benchmarked 66 different feature selection algorithms [42] and was confirmed by the studies shown in

Paper B. Study [42] also shows a brute force search for the optimal feature number to be best suited to decide the number of features to select. Especially in combination with ensemble feature extraction and ranking, this puts strong attention on the computational cost of the following processing steps that must be executed to swiftly evaluate the selected feature subsets.

4.7 Projections and Linear Discriminant Analysis

After feature selection, projections are one of the most common approaches to dimensionality reduction. They aim to show as much relevant information as possible with as few dimensions as possible. In contrast to feature extraction, which has similar goals, projections utilize features from different sensors and use the target vector to discriminate between relevant and irrelevant information. I.e., in the context of machine learning, projections perform information or sensor fusion and reduce the learning problem to its intrinsic dimensionality. As the algorithmic approaches of different projection algorithms are very diverse, giving a broad overview is left to respective review papers [95, 96].

However, as described in the previous section, an extremely cheap evaluation of feature subsets is needed since those evaluations require cross-validation of all the following steps. This need rules out algorithms like autoencoders that require iterative optimization procedures during training and algorithms like kernel-based projections that scale poorly even with few training samples. Ideally, the utilized algorithm allows analytical computation of the solution and scales linearly with the number of training samples. Linear Discriminant Analysis (LDA) fulfills those criteria [97]. Also, LDA and subsequent classification into the class with the lowest Mahalanobis distance [98, 99] to the group mean have been shown both in previous work [42] and in more recent benchmarks [100] to fall back only slightly behind SVM or KNN in terms of classification performance for condition monitoring tasks. Last but not least, after calculation, LDA can be represented by a single projection matrix of the size of the number of features times the number of groups minus one, which contributes to fulfilling memory constraints on smart sensors.

Given multiple groups with Gaussian-distributed samples with different mean values but identical covariance matrices, LDA computes a linear projection. This projection is optimal in maximizing between-class scattering (maximizes differences between groups) and simultaneously minimizes within-group scattering (maximizes within-

group similarities) [101]. In terms of measurement science, this is equivalent to finding the projection that provides optimal measurement contrast between groups. To do that, LDA maximizes the criterion function J(w) [101]:

$$J(\vec{w}) = \frac{\vec{w}^T S_b \vec{w}}{\vec{w}^T S_w \vec{w}}$$

With S_b representing the between class scattering and S_w the within class scattering. Both are computed as follows [101] :

$$S_b = \sum_{i=1}^{C} N_i (\vec{m}_i - \vec{m})(\vec{m}_i - \vec{m})^T$$

$$S_w = \sum_{i=1}^{C} \sum_{j=1}^{N_i} \left(\vec{x}_{ij} - \vec{m}_i\right)\left(\vec{x}_{ij} - \vec{m}_i\right)^T$$

In those formulae, C is the number of classes, N_i is the number of training samples in class i, $\vec{m}_i$ is the mean of class i, m is the dataset's overall mean and $\vec{x}_{ij}$ is the feature vector of point j in class i. The solution is typically computed analytically by solving the eigenvalue problem [101]

$$S_w^{-1} S_b \vec{w} = \vec{w}\Lambda, \qquad \Lambda = (\lambda_1^0, \dots, \lambda_k^0)$$

The eigenvectors of $S_w^{-1} S_b$ maximize $J(\vec{w})$. The resulting projection matrix w can be interpreted as $C - 1$ discriminant functions sorted by descending order of measurement contrast between classes. The eigenvalue problem also shows a numerical issue encountered with highly correlated features that might lead to an ill-conditioned or even singular and, therefore, un-invertible covariance matrix S_w. Further disadvantages are the regularly unfulfilled assumption of equally distributed groups and the absence of a regularization parameter favoring overfitting. However, in the context of preceding feature extraction and selection of a suitable number of features, the sensitivity to correlated features and the tendency to overfit favors the selection of a small number of features that, in turn, boosts the robustness and interpretability of the linear projection. The boost in robustness due to smaller feature sets is one of the possible explanations for the performance of LDA, which is comparable with SVM and KNN [42]. Therefore, the main issue to be considered is violating the equal distribution assumption. Although it does not seem to lead to the problems in the evaluations performed in Papers 1 and 2 and Papers A-E, it harbors the risk of suboptimal error rates that originate in the LDA's tendency to project multiple sub-groups within a group onto one another to minimize within-group scattering. However, separate sub-clusters would provide better (non-linear)

separability. Ultimately, the advantages of such non-linear separability need to be traded for low computational cost during training and inference.

4.8 Classification and Regression

The following data processing step is classification or quantification using classification or regression algorithms to derive decision rules from the projected features. As for projections, low computational complexity in the number of features, low computational cost, and few needed parameters are desired algorithmic features deduced from the possibly high number of training samples, the need for many cross-validated evaluations, and the inference on smart sensors, respectively.

Classification:

As LDA aims to project each group onto a single cluster, classification into the group with the nearest center is a straightforward solution that requires minimal training effort. It only comprises calculating the group means and storing a neglectable number of parameters. Those parameters are the C group means in the low dimensional discriminant function space, i.e., $C * n$ parameters. Mahalanobis distance is standing to reason as a distance measure to account for the common violation of the LDA's equal covariance assumption [98, 102, 103]. Since the covariance matrices of all groups needed for Mahalanobis classification have to be computed for the LDA projection, the training procedure comes down to their projection onto the discriminant functions and their subsequent inversion. This procedure only adds a term proportional to the number of dimensions in feature space to the computational cost due to the fixed number of groups and discriminant functions. Therefore, this approach is suggested for condition monitoring in Paper 1.

Quantification

Alternatively, as shown in [25], the discriminant functions can be used for quantification and are suggested as such in Paper 1. This approach is based on the idea that the first discriminant function is designed to show the best group separation. That would be the direction of increasing wear down if suitably chosen quantification steps of wear are used as class labels. Therefore, the DF only needs rescaling and offset correction to be mapped to the numeric target value. In the context of condition monitoring, this approach offers multiple advantages and additional insights not provided by the application of common regression methods:

- Unlike linear regression algorithms, LDA does not assume a linear correlation between wear progress and wear symptoms and still offers inherent and global explainability.
- Visualizing discriminant functions as a scatter plot gives an intuitive insight into the extent of nonlinearity between wear progress and wear symptoms. This insight is gained in two ways. First, LDA would show groups with more significant changes in wear symptoms farther apart or with a greater scattering in wear direction as groups in areas with minor changes. Second, a linear correlation between wear progress and symptoms could be shown entirely on the first DF. However, a nonlinear correlation would require representing information on higher DFs orthogonal to the first. Therefore, the amount of information on the first DF could be a measure of linearity.
- Similarly, LDA shows nonlinearities, jumps, and other influences from perturbance variables, which cannot be linearly compensated for by LDA (see Paper 1). Such jumps in wear symptoms typically occur on breakouts on cutting edges of drills and milling tools.
- Lastly, LDA can provide an implicit plausibility check for the learned model. As the LDA is unaware of the natural order of groups given by increasing wear-down, showing this order in a projection on the DFs proves that LDA could rediscover this order from the wear patterns, which supports their causal character. An example can be seen in Figure 5.

These advantages mainly apply during model building and explorative data analysis. LDA can be replaced if needed by a suitable regression algorithm for deployment. This algorithm must offer low computational cost, compensate for the nonlinear effects shown by LDA, or be used with a revised target value with a linear regression algorithm like partial least squares regression [104].

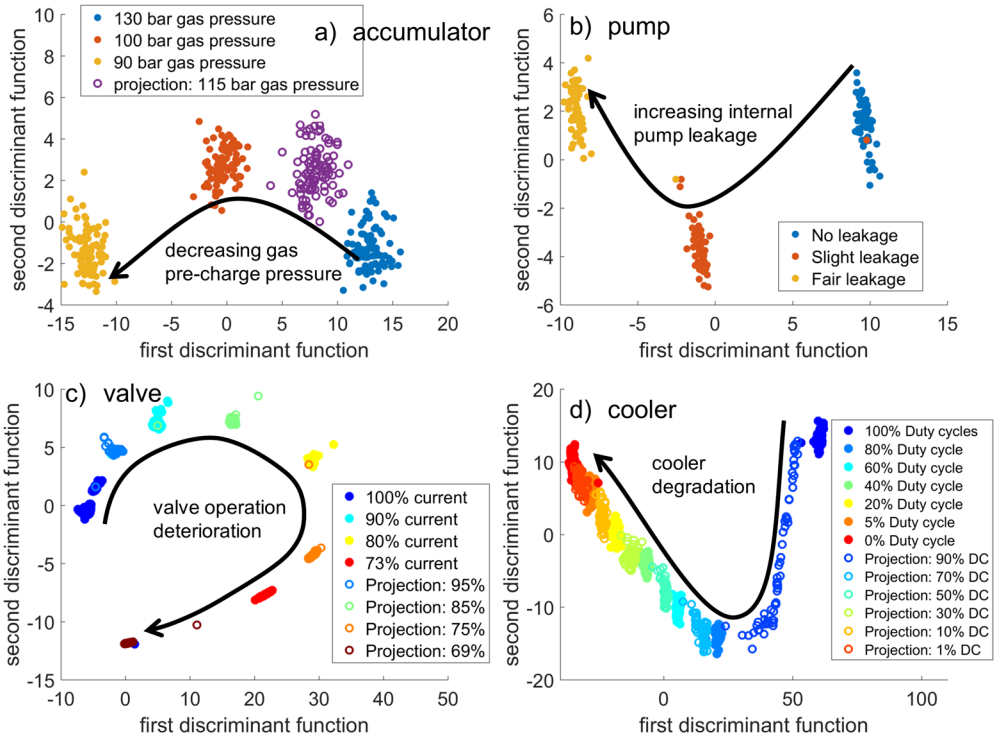


Figure 5: Results for determination of the four system faults studied: accumulator pressure (a), internal pump leakage (b), valve operation (c), and cooler degradation (d). Full symbols show data used for determining the statistical model, and open symbols show additional test data not used in the training, which proves that unknown data are interpreted correctly [Paper 2]

4.9 Validation and Test

Validation is an essential step of machine learning and tries to estimate the expected performance of an algorithm after deployment. Although there are multiple different methods for validation, they are all based on the same principle of out-of-sample testing. All methods divide the available data into at least two sets, of which one is used for training the algorithm and the other to test and report its performance on previously unseen data.

If hyperparameters of the trained algorithm are tuned to fit the data or the best algorithm is to be selected from a given set of algorithms, a second split on the available data is required to estimate the expected performance of said selection. The resulting datasets are training, validation, and test data. Thereby, validation data is used to tune hyperparameters or to select the algorithm, and test data is used to evaluate the final performance.

Based on how the data is split, the validation tests for different kinds of robustness, e.g., random cross-validation tests for statistical significance and robustness against random noise. As shown in Paper B, domain shifts are the most dominant influences

on deployed performance in the context of smart sensors. Therefore, the employed validation procedure has to test robustness against domain shifts by splitting the data into sets that exclusively contain data from a single domain, which could be the same component, the same machine, or the same operational setting of a machine. This procedure is known as leave one group out and can also be used for interpolation tests, as shown in Figure 5.

Since leave one group out cross-validation is not always possible, k-fold cross-validation is the most common validation technique used in literature. It splits the available data into k, usually randomly drawn groups, trains with k-1 groups, and reports performance on the left-out group while iterating over the different groups to be the test group. Usually, while samples are randomly assigned to each group, the assignment algorithm preserves the overall proportion of the different classes in every group (stratified sampling).

Both random and group-based cross-validation can be used for both validation and testing. However, a single fixed test set is usually chosen by random selection (k-fold cross-validation) or group (group-based cross-validation) to limit the computational cost of nested cross-validation. For group-based cross-validation, the test group is usually chosen to pose an interpolation problem when training is performed with the remaining groups. However, even when done correctly, error propagation in measurement science, according to GUM, as shown in Paper C, is still preferred over cross-validation.

4.10 Edge-AI

Edge AI currently focuses on the inference of machine learning models or feature extraction. The training of those models is usually done offline using open-source frameworks like "TensorFlow" [105]. It utilizes already available accelerators, e.g., from NVIDIA, Intel (GPU-based), or Xilinx (FPGA-based). However, those accelerators' high energy consumption and cost prohibit their use in most smart sensor applications.

For microcomputers (edge-computing), early software solutions for model inference like "TensorFlow lite" [106] are available to optimize and deploy models on the edge. Also, the first examples of size-limited neural networks that utilize embedded AI to detect, e.g., operating modes from current measurements, have been shown [107].

The most promising concepts for embedded AI are:

- Classical microcontroller with specially adopted networks [108].
- Microcontroller with hardware accelerator [109].
- Neuromorphic structures (in early research) [110].

Many already available AI accelerators are based on hardware-accelerated matrix multiplication (e.g., NVIDIA TensorCore [111]) with reduced computational accuracy in comparison with conventional CPUs and GPUs (see Figure 6). To what extent this reduced computational accuracy, motivated by image recognition with a maximum of eight bits per color channel, can be transferred to other applications like audio-based fault detection with 16-24-bit measurement resolution is an open research question.

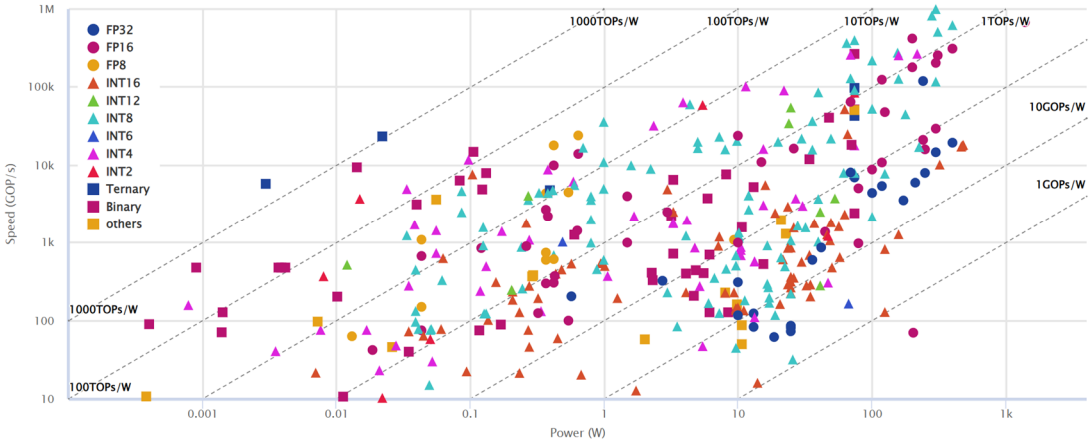


Figure 6: Currently available AI accelerators and their computational accuracy [80].

Most currently available accelerators target applications in image, video (e.g., Intel Movidius is marketed as a Visual Processing Unit [112]), and natural language processing. This focus is due to the massive amount of training data available in those domains, favoring the huge success of machine learning and accelerators' advantages, especially in those applications.

The hardware accelerators are usually shipped with software frameworks for neural network inference. However, it is possible to use them to accelerate other algorithms like linear transformations naturally supported by accelerators based on matrix multiplication [111] or Support Vector Machines [113].

Accelerators optimized for industrial applications that implement the approach suggested in Papers 1 and 2 are currently under development [30, 31]. Simultaneously, neural network representations of algorithms employed for industrial automatic machine learning are being researched as an alternative solution that utilizes generic neural network accelerators [114].

5 Automated Machine Learning for Condition Monitoring

5.1 Paper A: Sensors 4.0 – Smart Sensors and Measurement Technology Enable Industry 4.0

This paper highlights the vision of Sensors 4.0 introduced in Section 3. It puts it into the context of an ongoing sensor evolution from simple indicators to fully integrated smart sensors with communication capabilities. It identifies measurement as a service, traceability of individual components, self-learning systems, and semantic technologies as major trends in sensor technology. It then focuses on Condition monitoring using data-based modeling as a new sensing paradigm with examples for monitoring machine and sensor faults. This again motivates the research in Papers 1-3 and shows its significance in measurement science.

Another contribution of this paper is the demonstration of similarities between classical measurements and statistical model-based prediction. Namely, model predictions exert characteristics similar to physical sensors, like constant uncertainty over a pre-calibrated measurement range or temperature dependency of that uncertainty. Last, the paper motivates using a modular algorithmic approach to machine learning that will be widely adopted in Papers 1-3.

Sensors 4.0 – Smart Sensors and Measurement Technology Enable Industry 4.0

Andreas Schütze[1,2], Nikolai Helwig[2], and Tizian Schneider[2]

[1]*Saarland University, Lab for Measurement Technology, Saarbrücken, Germany*

[2]*Centre for Mechatronics and Automation Technology (ZeMA gGmbH), Saarbruecken, Germany*

Journal of Sensors and Sensor Systems (2018), 7, 359-371

The original paper can be found online at https://doi.org/10.5194/jsss-7-359-2018.

© 2018 by the authors. This open-access article is distributed under the terms and conditions of the Creative Commons Attribution 4.0 License. (http://creativecommons.org/licenses/by/4.0/).

J. Sens. Sens. Syst., 7, 359–371, 2018
https://doi.org/10.5194/jsss-7-359-2018
© Author(s) 2018. This work is distributed under
the Creative Commons Attribution 4.0 License.

JSSS Journal of Sensors and Sensor Systems Open Access

Sensors 4.0 – smart sensors and measurement technology enable Industry 4.0

Andreas Schütze[1,2], **Nikolai Helwig**[2], **and Tizian Schneider**[2]

[1]Lab for Measurement Technology, Department Systems Engineering, Saarland University, 66123 Saarbruecken, Germany
[2]Centre for Mechatronics and Automation Technology (ZeMA gGmbH), 66121 Saarbruecken, Germany

Correspondence: Andreas Schütze (schuetze@lmt.uni-saarland.de)

Received: 13 November 2017 – Accepted: 25 February 2018 – Published: 9 May 2018

Abstract. "Industrie 4.0" or the Industrial Internet of Things (IIoT) are two terms for the current (r)evolution seen in industrial automation and control. Everything is getting smarter and data generated at all levels of the production process are used to improve product quality, flexibility, and productivity. This would not be possible without smart sensors, which generate the data and allow further functionality from self-monitoring and self-configuration to condition monitoring of complex processes. In analogy to Industry 4.0, the development of sensors has undergone distinctive stages culminating in today's smart sensors or "Sensor 4.0". This paper briefly reviews the development of sensor technology over the last 2 centuries, highlights some of the potential that can be achieved with smart sensors and data evaluation, and discusses success requirements for future developments. In addition to magnetic sensor technologies which allow self-test and self-calibration and can contribute to many applications due to their wide spectrum of measured quantities, the paper discusses condition monitoring as a primary paradigm for introducing smart sensors and data analysis in manufacturing processes based on two projects performed in our group.

1 Introduction

"Industrie 4.0", branded the fourth industrial revolution, is in fact more of a political vision than a new technical paradigm (see *Plattform Industrie 4.0*, 2018; *Hightech-Strategie: Industrie 4.0*, 2018; *Dossier: Digitale Transformation in der Industrie*, 2018): it is simply the continuing progression of achieving better knowledge and control over the entire production process that has been ongoing since industrialization made efficient mass production possible. The main benefit of this new way of looking at things is the chance to establish new business models. This is actually expressed better by the Anglo-Saxon term Industrial Internet of Things or IIot (Industrial Internet Consortium, 2018) because it hints at transferring successful business models of the new economy to industrial application; even more indicative are terms like digitalization or, short and pithy, Googlification. In fact, other application fields especially in consumer services are far ahead of industrial production processes in making use of the power of digitalization. Note that this is actually one fundamental difference between the common approach and the new thinking: services and hence the value to the customer are more important than products.

This paper addresses the importance of sensors, instrumentation, and measurement science for Industry 4.0 and discusses potential and trends; it is based on two conference presentations addressing smart sensors, their relevance for Industry 4.0, as well as the requirement for an expanded uncertainty evaluation (Schütze, 2015; Schütze and Helwig, 2017); a shorter German version was recently published elsewhere (Schütze and Helwig, 2016). In the field of sensors and instrumentation the trend towards smart sensors has long been established in aspects like better performance, higher integration, and multi-parameter sensing, but also built-in intelligence as well as secure and safe networking (Gassmann and Kottmann, 2002; Sensor-Trends, 2014). Intelligent sensor systems allow e.g. self-identification or diagnosis up to self-configuration, calibration, and repair, often subsumed under the term self-X (Akmal Johar and König, 2011). In

analogy to Industry 4.0 the term Sensor 4.0, coined by Peter Krause, the chairman of AMA e.V. (AMA, 2018), snappily designates the current development in sensors and measurement science. Similar to the four phases of industrial development, this classification discriminates between purely mechanical indicators (e.g. the aneroid barometer, also referred to as a Vidie can, invented by the French physicist L. Vidie in 1844), electrical sensors (e.g. classic strain gauges invented simultaneously by E. E. Simmons at Caltech and A. C. Ruge at MIT in 1937/38), the state-of-the-art electronic sensors (e.g. electronically compensated pressure sensors available since approx. 1970), and, finally, smart sensors; cf. Fig. 1. This overview also shows how strongly the industrial evolution is connected to sensors and instrumentation. Note that the importance of sensors is not limited to industrial processes, but is in fact strongly influencing all current megatrends like smart cities or smart mobility. The best examples of highly integrated sensor platforms are in fact smartphones which typically integrate around 15 different sensors and heavily make use of multisensory signal evaluation, e.g. for navigation which relies on accelerometers, gyroscopes, magnetometers, and pressure sensors. At the same time these sensors are also used for other services like weather monitoring, screen orientation, step counting, and, last but not least, gaming. In this case, the sensors are in fact "dumb" sensors, with the integration and data fusion between different sensors achieving a smart platform.

The relevance of modern sensors and instrumentation is also reflected by the economic data which show a continuous growth of more than 6 % CAGR in turnover from 2005 to 2015 and a steady increase in jobs of almost 40 % over the same period, compared to stagnation in the industry as a whole (based on an analysis by AMA). The companies in this field also invest an average of 10 % of their turnover in research and development and are therefore attractive employers for young engineers and physicists.

2 State-of-the-art and current trends

A current trend in sensor technologies is the use of active measurement principles that are used in various sensor elements and systems. Examples are magnetic sensors, e.g. Hall sensors using spinning current (Munter, 1990), internal calibration and even correction of offset temperature coefficients through integration of internal chip heaters (Stahl-Offergeld, 2011), MR sensors using the compensation principle to suppress temperature cross-sensitivity (Marien and Schütze, 2009), micromechanical accelerometers (also using compensation or internal calibration methods) and gyroscopes (using the Coriolis effect with active vibration excitation), and Coriolis-based flow sensors or chemical sensors using temperature modulation for improved selectivity, sensitivity, and stability (Reimann and Schütze, 2014; Baur et al., 2015). Active modes of operation also offer additional potential for self-diagnosis, which is already extensively being used in automotive applications (Ochs, 2013). This does not only apply to inertial sensors, where the correct function is checked with internal excitation, but also to e.g. the lambda probe: here the time constant for heating to the desired operating temperature is used to detect faults, e.g. cracks of the ceramic. Self-diagnosis is especially important for applications in safety and security. Fire detection and explosion protection could not be addressed with low-cost gas sensors, which are prone to poisoning. Here, dynamic operation also allows detection of sensor faults, e.g. poisoning of the sensor material (Bastuck et al., 2015; Schüler et al., 2015).

Magnetic sensors are especially suitable for self-X methods as an internal calibration can be realized by coils integrated in the system or directly on the chip. Furthermore, as the sensors are very small and integrated on silicon chips with good thermal conductance, heating of the sensors is also possible, thus allowing direct determination of thermal cross-sensitivity. Many principles are today already implemented in integrated Hall sensors (e.g. 3-D-HallinOne sensors developed by Fraunhofer IIS), due to their being based on standard CMOS technology, thus allowing simple integration with analogue and digital electronics (Stahl-Offergeld, 2011). MR sensors are not yet as advanced due to the difference in technologies for sensor chips and electronics, but the potential for self-X technologies is increasingly being studied (Akmal Johar and König, 2011; MoSeS-Pro, 2015). Note that future trends might include internal traceability of the sensor function by making use of quantum standards for SI units so that sensors might be truly calibrated during operation as proposed by Kitching et al. (2016) (see also NIST-on-a-Chip, 2018).

Integrated Hall sensors can serve as one specific example highlighting the potential of (magnetic) sensors and their integration with advanced modes of operation and data treatment in the sensor itself. Hall sensors are used in many applications and are sold in large quantities at surprisingly low cost considering their performance. While on the outside these sensors still resemble the well-known simple Hall plate, a purely analogue, current, or voltage driven sensor with voltage output to measure the magnetic field, they are much more complicated inside. The spinning current principle, periodically switching driving and output contacts, has already been used for a long time (Munter, 1990) to compensate for various unwanted aspects (unsymmetrical geometry, variations in doping of the Hall layer, mechanical strain, and temperature differences), which would otherwise result in large offsets and therefore reduced resolution (Stahl-Offergeld, 2011). Even after spinning current compensation which is achieved by typically four measurements with current induced in all four directions of the Hall plate and subsequent averaging of the results, a residual, temperature-dependent offset remains. By integrating a small excitation coil directly on the chip this offset can be determined during normal operation. In addition, a small heater can also be

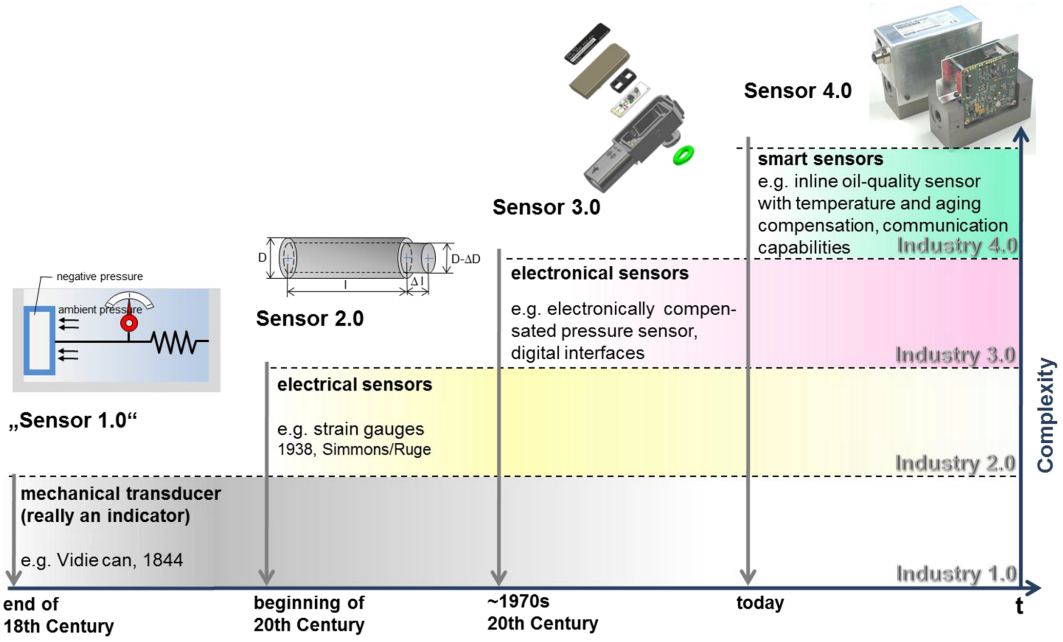


Figure 1. Historic evolution from "Sensor 1.0" (without electrical output this is not a sensor according to the usual definition) to smart sensors, i.e. "Sensor 4.0" (based on Peter Krause, chairman of AMA e.V.).

integrated on the chip, which can induce small temperature changes to determine the temperature coefficient (TC) of the offset, thus allowing a digital correction of the temperature-dependent offset with a linear model. To achieve offset compensation over a wide operating temperature range, this procedure can be repeated at different ambient temperatures. The resulting residual offset and thus the effective resolution of the sensor is greatly improved by this procedure by more than 1 order of magnitude from ±1 mT down to ±50 µT (Stahl-Offergeld et al., 2009; Stahl-Offergeld, 2011). However, due to cost restraints the sensor only contains one AD converter, which means that chip temperature and Hall voltage cannot be measured simultaneously. To avoid errors in the determination of the offset TC the thermal behaviour of the chip is therefore modelled with several time constants. Furthermore, several measurements during this self-calibration and a statistical evaluation of the resulting variations are used to check whether the applied external field has changed during this procedure, which could lead to false values for offset and offset TC; for further details, the reader is referred to Stahl-Offergeld et al. (2009). In addition, other production-related parameters of the sensor chip can be determined with a suitable strategy combining on-chip measurements and digital evaluation (Stahl-Offergeld et al., 2010; Stahl-Offergeld, 2011). This example shows that even a seemingly simple Hall sensor today determines the required measurement value based on a complex digital process. Note that a strict determination of the resulting measurement uncertainty based on the GUM principles (GUM, 2008) would result in a very complex process if the complete system and a full physical model were to be considered. In this case, a suitable statistical approach for determination of the uncertainty would seem more suitable.

Thus, smart sensors with additional functionality provide a significant added value for higher-level functions, e.g. in production systems. The correct sensor function is also required for the condition monitoring of complex systems (see Sect. 3 below). In this case, the correlation of sensor data within the system can also be used to verify the correct sensor function; however, in this case, the sensor fault diagnosis has to be performed on a higher level within the system.

Additional trends that will be initiated or at least pushed further by the Industry 4.0 paradigm are the following.

- Measurement as a service: this could be a trend similar to the service provided by Uber in public transport, i.e. measurement services or even individual results are sold instead of instruments. Note that the measurement uncertainty – determined online by self-calibration – will then influence the price.

- Traceability of individual components down to screws, individual gears and even gaskets: this additional knowledge will allow tolerance measurement in the assembly of (sub-)systems and is also required for a comprehensive condition monitoring to assess the influence of individual processing steps and machines on the final result.

- Self-learning systems: the correlation between sensor data as well as other process and ambient parameters

can be evaluated to ensure the correct function of the system in the sense of a system self-diagnosis by making use of machine learning (Cachay and Abele, 2012). So far it is unclear whether unsupervised methods are sufficient or whether supervised learning, see Sect. 3 below, is required, i.e. knowledge of the current system status for training the evaluation.

– Semantic technologies for analysis of complex systems: interpretation of measurement values beyond the purely data-based approaches could offer further opportunities, e.g. for plausibility checks of sensor data and for providing confidence values for (fault) causes. Note that the World Wide Web consortium (W3C) started working on a semantic sensor network ontology as early as 2005 which allows representation of measurement values and their significance (Semantic Sensor Network Ontology, 2017).

The last example shows that the importance of sensors and measurement technology was recognized also by other parties, which leads to some parallel and independent developments. Interestingly, however, aspects like measurement uncertainty and sensor self-monitoring are not addressed in the context of semantic technologies even though semantic representation would be highly valuable especially for these aspects.

3 New measurement paradigm: condition monitoring using data-based modelling

The potential of data-based sensor signal evaluation is demonstrated by the iCM Hydraulics project (2013). In this project a hydraulic model system combining a primary circuit with variable load and a secondary circuit for cooling and filtration were used to study the identification of typical system faults (internal pump leakage, delayed valve switching, pressure leakage in the accumulator, reduced cooling efficiency) only based on an evaluation of the usual process sensors (pressure flow rate, temperature, electrical power).

Figure 2 provides an overview of the approach: the hydraulic system is equipped with a total of 17 physical and virtual (e.g. efficiency calculated from electrical power input and hydraulic power output) sensors, which are read out with up to 100 Hz. The system was used to simulate a periodic industrial process with a work cycle of 1 min duration. In each cycle a total of approx. 50 000 raw values is recorded, which are interpreted as a high-dimensional measurement vector. A multi-step dimensionality reduction covering signal pre-processing, feature extraction, and selection yields a projection obtained by linear discriminant analysis (LDA) (Duda et al., 2000), which allows classification of the system status, i.e. identification and quantification of the fault. Classification can be performed with various methods, e.g. k-nearest neighbours, support vector machines (SVMs), or artificial neural networks (ANNs). Note that pre-processing and feature extraction are realized with unsupervised methods, i.e. without making use of the system status, while feature selection – here based primarily on Pearson correlation of features and fault status – and LDA projection are supervised methods, i.e. require the knowledge of the system status (Helwig and Schütze, 2014). The evaluation is based on a comprehensive training phase in which all combinations of all fault states are measured. The complete training is based on several thousand working cycles and requires approx. 3 days, primarily due to the relatively slow equilibration of the temperature after changing the cooling efficiency. The complete training data set contains almost 120 million raw data points. A systematic validation, e.g. based on k-fold cross-validation, completes the development of the statistical model and ensures that no overfitting occurs in spite of the high-dimensional input data set and the supervised training methods (Helwig and Schütze, 2014).

In this example statistical methods were primarily used for feature extraction. The working cycle was divided into 13 sections (complete cycle and 12 sections representing different constant or changing pressure levels, respectively) and the first four statistical moments (mean, standard deviation, skewness, and kurtosis) were determined for each sensor in each section. This can be implemented on low-cost hardware very efficiently, but is still the computationally most costly step of the training procedure. This step requires a few minutes on a standard PC for the complete data set with several 1000 cycles. The resulting almost 900 features (17 sensors $\cdot$ 13 sections $\cdot$ 4 statistical moments) result in a feature space that still has too many dimensions for efficient classification. Therefore, feature selection based on correlation between features and target classes, i.e. fault level, is used which is computationally extremely efficient, same as the calculation of the LDA projection to obtain the 2-D plots, cf. Fig. 3, or ideally only one discriminant function (DF) per system fault. These two computation steps only require fractions of a second. Even faster is the classification of a new working cycle, i.e. extraction of the selected features, projection in the LDA space for each system fault, and classification based on a k-nearest neighbour classifier, which can thus be performed in real time even on a low-cost microcontroller-based system.

The performance of the approach is shown in Fig. 3 for the four studied system faults: each fault state can be identified independently and its severity or level can be estimated with surprisingly high accuracy. The cooler efficiency, for example, can be estimated with better than 10 % (the reduced cooler efficiency was simulated with pulse width modulation of the power supply, and the percentage gives the duty cycle used); the accumulator pressure can be determined with an uncertainty of approx. 5 bar. Projected test data which were not used to build the model (open symbols) show that the model allows correct classification of unknown states and even that an extrapolation of data outside the training range is possible within limits.

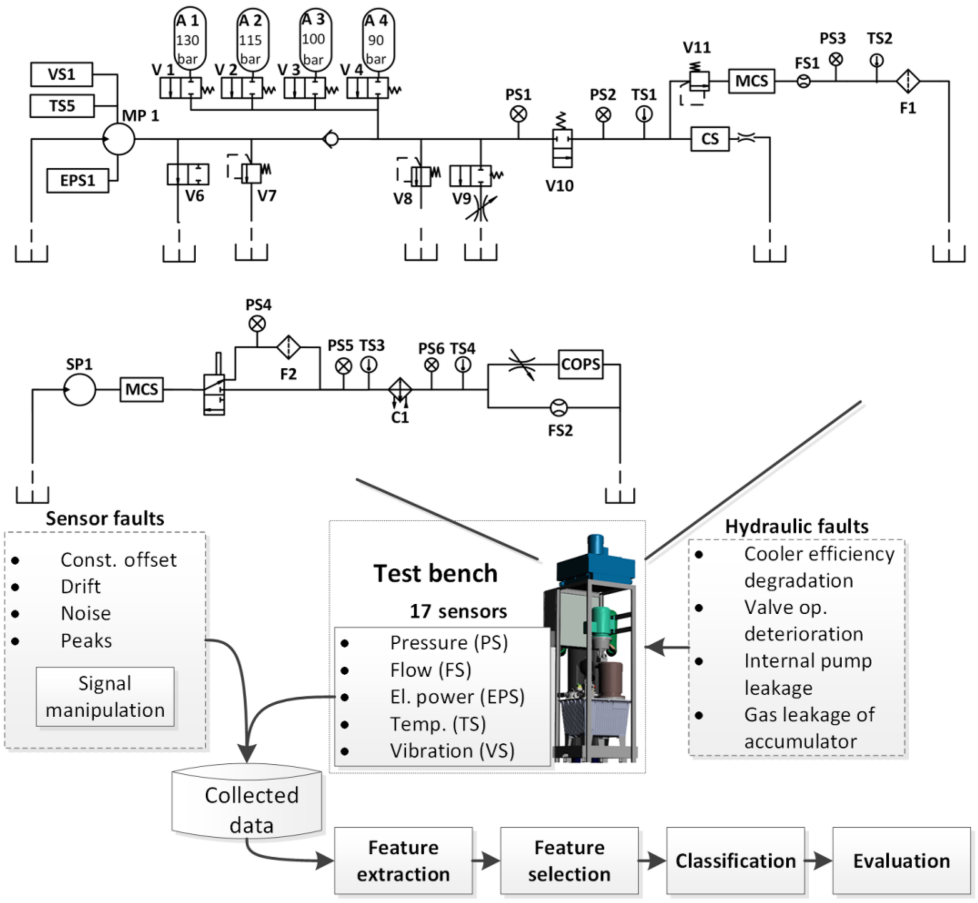


Figure 2. Hydraulic system and concept for the data analysis within the iCM-Hydraulics project (Helwig and Schütze, 2015). A statistical model was built allowing for independent identification of various hydraulic system faults as well as identification of sensor defects. System condition monitoring remained possible even with sensor faults by excluding the identified faulty sensors from the analysis.

In further experiments we have shown that the training can be transferred from one system to a second, identical system after some calibration, i.e. shift of the LDA projections for the correct system state (Helwig et al., 2015a). Given the high performance which was not expected when designing the experiments, we also studied how sensor faults would influence the classification results. For this, sensor offset, drift, noise and signal drop-outs were simulated in the recorded data for all sensor channels and the resulting data were used to classify the system state. Not surprisingly, the classification rate is drastically reduced, especially for monitoring of pump leakage and the hydraulic accumulator. To allow automatic recognition of sensor faults, these were defined as new targets for the classification algorithm and trained using the same completely automated approach. Again, the simulated sensor faults could be recognized with high reliability independent of the system state as shown in Fig. 4 for two exemplary sensor faults. In fact, sensor faults can be diagnosed before they lead to false classification of the system state (Helwig and Schütze, 2015). Correct classification of the overall system state is still possible by excluding the defective sensor(s) from the evaluation and making use of the remaining sensors. In fact, up to five of the most important sensors can be excluded from the evaluation and still a correct classification rate of more than 80 % is achieved (Helwig and Schütze, 2015).

The projection shown in Fig. 4a can also be presented in a different way, as the second discriminant function (DF2) obviously does not provide relevant information for the offset classification. Plotting the data as a histogram results in the plot shown in Fig. 5 which shows nearly normal distributions for all six classes with a constant FWHM (full width at half maximum) or standard deviation. Thus, this projection could be used not only to determine or "measure" the sensor offset; it also provides an estimate for the uncertainty with which this offset can be determined, considering only type A uncertainties. Note that this also holds for the two classes with 2 and 10 bar offset, which were not used for calculating the LDA, i.e. building the statistical model.

Similarly, the histogram for the accumulator pressure shown in Fig. 6 also yields a constant standard deviation, which increases with increasing temperature range, thus in-

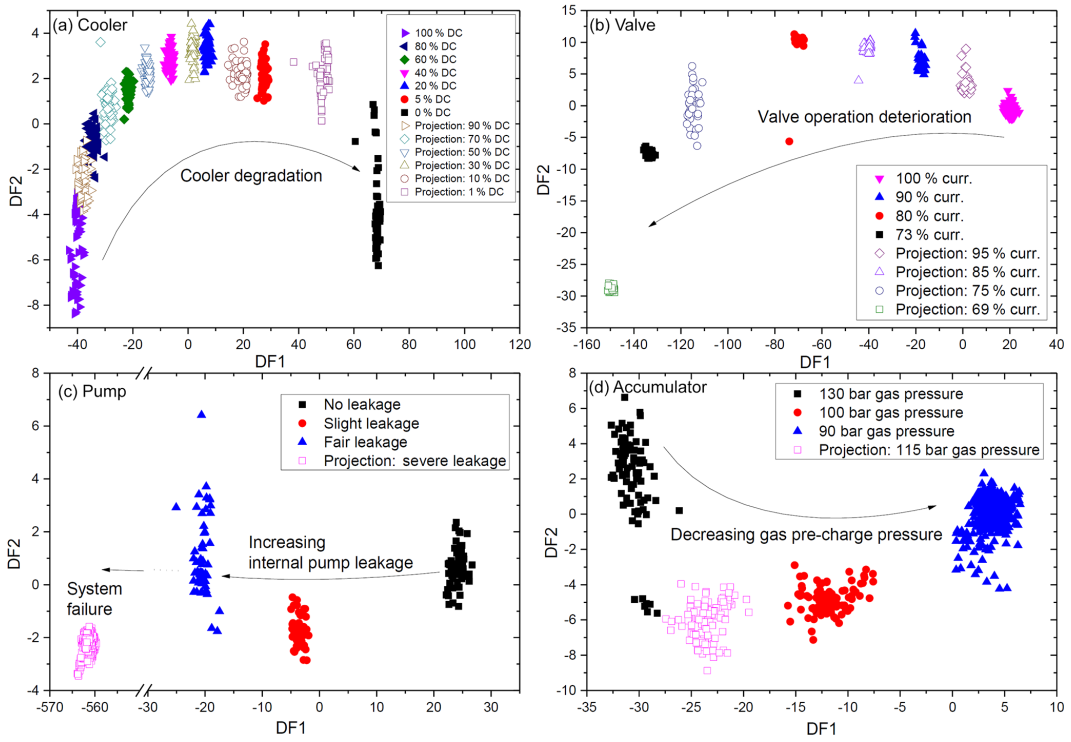


Figure 3. Results for determination of the four system faults studied: cooler degradation **(a)**, valve operation **(b)**, internal pump leakage **(c)**, and accumulator pressure **(d)**. Full symbols show data used for determining the statistical model, and open symbols show additional test data not used in the training which prove that unknown data are interpreted correctly (Helwig and Schütze, 2015).

dicating a measurement uncertainty depending on ambient conditions, which is quite common for many sensors. This also shows that the performance of the condition monitoring approach deteriorates drastically when the temperature of the system, or more specifically the temperature of the hydraulic oil and with it the viscosity, changes over a wide range. While the accumulator pressure can be estimated with an uncertainty of less than ± 5 bar for a temperature range of 10 °C, an increase in the temperature range to 20 °C leads to significant overlap between the different classes with an uncertainty of at least ± 10 bar. Note that narrowing the temperature range further does not reduce the uncertainty correspondingly, probably due to noise of the sensor data contributing to this result (remember that an discriminant function is a weighted sum of different features, i.e. sensor values). To take this effect into account the training of the statistical model would either need to be extended to include data over a wide (oil) temperature range or the exact interpretation of the system condition can only be done in a typical operating window. The latter approach is surely better suited for typical industrial applications, especially as a full condition monitoring is not required with high temporal resolution, i.e. for classification of wear processes, due to the normally slow progression of the system deterioration. On the other hand, if this approach were to be used for mobile (hydraulic) machinery, i.e. loaders, the ambient and also the operating temperatures would depend drastically on location and weather conditions. In this case, either an expanded training over the full operating temperature range would be required or perhaps several different projections selected based on the relevant temperature level. In any case, training effort would increase to allow universal condition monitoring.

The examples shown here clearly demonstrate the potential of data-based statistical modelling for condition monitoring of complex systems purely based on existing process sensors. Thus, a cost-efficient and powerful monitoring can be achieved which allows interpretation of the results also in terms of the measurement uncertainty of the systems status, i.e. the uncertainty is nearly constant over the full range from a system in mint condition to near failure, but the uncertainty increases if additional factors, in this case significant changes in the oil temperature, have to be taken into account. Note, however, that this does not apply to all system faults. In this example a varying uncertainty was observed for the valve switching behaviour which increased over the monitored range, which might be due to a non-linear relation between features and resulting discriminant function and the fault status. Even more problematic is the observation that the variation of results for test data does not show a normal distribution, i.e. a simple interpretation of the standard deviation as measurement uncertainty is not possible, and,

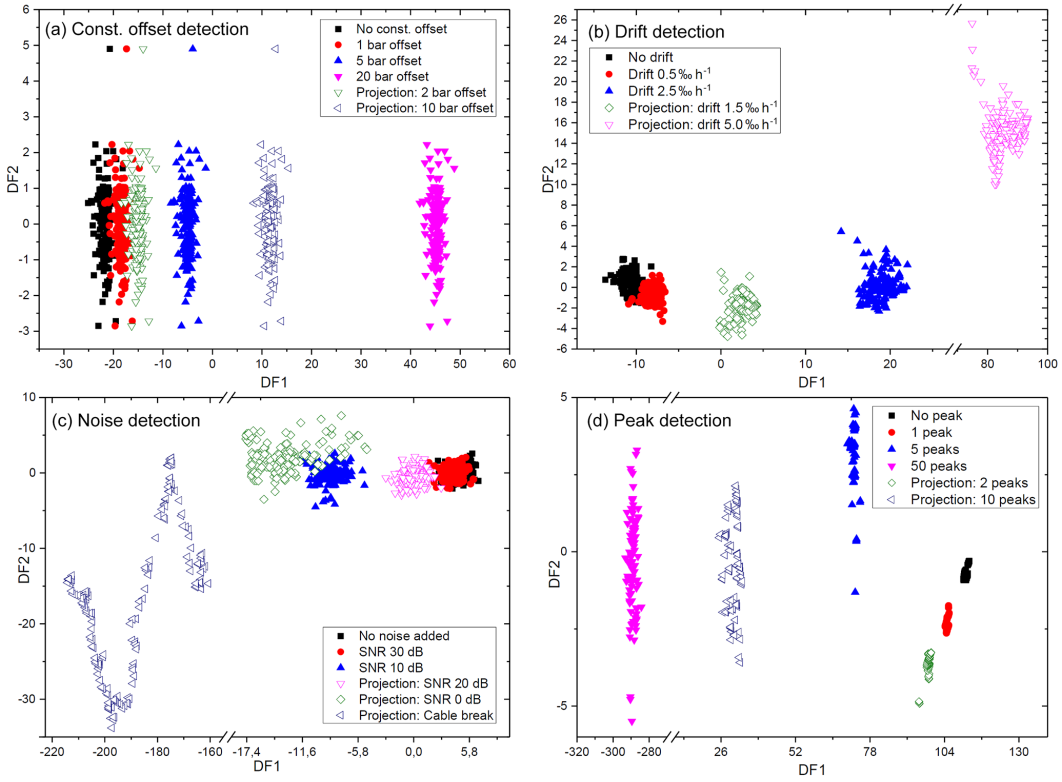


Figure 4. Results for identification of sensor faults offset **(a)**, drift **(b)**, noise **(c)** and peaks (signal drop-outs, **d**). Full symbols show data used in determining the model for the sensor faults diagnosis, and open symbols show additional data not used in the training, which again proves that unknown data are interpreted correctly (Helwig and Schütze, 2015).

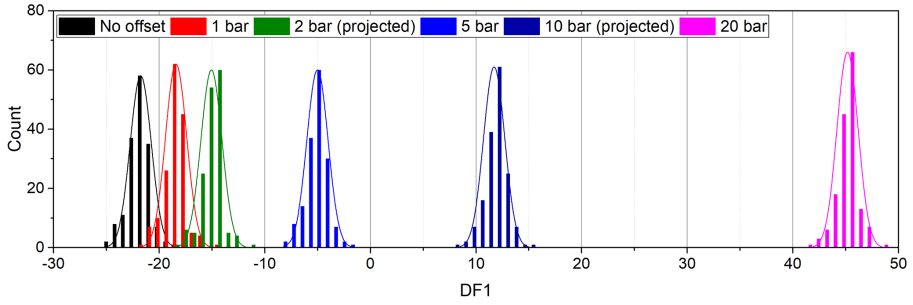


Figure 5. Plotting the data from the LDA analysis as a histogram of the first discriminant function (DF) shows that this projection results in a linear relationship allowing determination of the sensor offset. An estimate of the uncertainty for the offset is also possible due to the constant standard deviation of the data; note that this includes data at 2 and 10 bar offset, which were not used for building the statistical model.

furthermore, the interpretation of the statistical results as a "measurement" of the system state might not be justified.

4 A modular approach for smart sensor networks and condition monitoring

The successful preliminary work in iCM Hydraulics resulted in the establishment of a successor project, in which the developed methods are transferred to an open sensor system toolbox. In this project (MoSeS-Pro, 2015) magnetoresistive sensors (AMR, GMR, and especially TMR) are primarily used to measure current, position, and angle, but other (micro)sensors, e.g. MEMS sensors for vibration, pressure, or thermal radiation, are also used to extend the measurement spectrum. These sensors are also integrated into components and subsystems (Helwig et al., 2017b) to allow improved performance and condition monitoring, both as an end-of-line test in their production and during their opera-

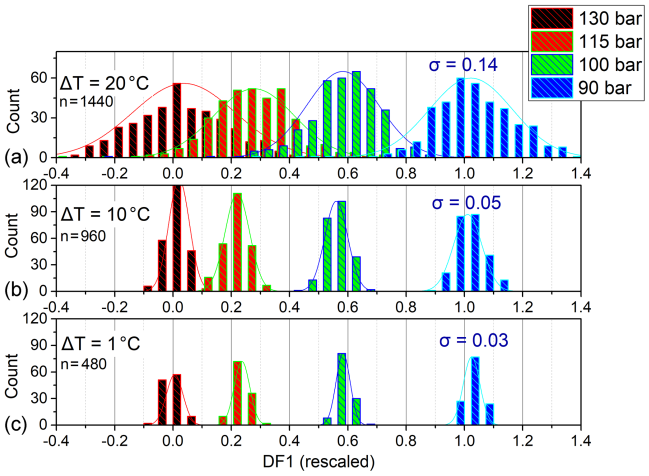


Figure 6. Histogram of the sensor data/features for classification of the accumulator pressure vs. first discriminant function. Here, the data were rescaled (compare Fig. 3d) to centre values for 130 bar (nominal pressure) at 0 and data for 90 bar at 1. The standard deviation for small temperature range **(c)** seems to be determined by noise from the sensors/features, while it is strongly influenced by variations of the oil temperature for increasing temperature range **(a)**.

tion in manufacturing systems. In this project, modular electronics and software algorithms are developed, allowing the required signal pre-processing and feature extraction directly in the smart sensor. Otherwise, signals recorded at high frequencies of several 100 MHz would result in data rates which would overload the higher levels. In addition, novel self-X methods, wireless sensor interfaces, and energy harvesting are developed for easy integration and initialization of system operation. Figure 7 gives an overview of the modular approach.

As shown above, statistical data analysis is a powerful tool for condition and process assessment without firm and detailed expert knowledge since most of the underlying algorithms are self-optimizing and can be concentrated in automated signal processing chains. However, this approach, especially in the case of supervised learning, requires a sufficient quality of training data, i.e. typically cyclical process-synchronized sensor data which are annotated with corresponding classes, i.e. the target vector for which the statistical model is to be trained. The typical steps for offline analysis (Fig. 8a) are signal pre-processing, feature extraction, and selection as well as classification with subsequent evaluation and can be interpreted as a gradual dimensionality reduction. Feature extraction and selection can be fully automatized using a modular approach based on complementary algorithms to extract information from the time domain, i.e. with adaptive linear approximation (ALA), from the frequency domain, i.e. with Fourier analysis, from the time–frequency domain, i.e. using wavelet analysis, or the overall system, i.e. based on principal component analysis (PCA). Similarly, complementary techniques are used to select suitable features and feature combinations, i.e. simple correlation analysis or recursive feature elimination support vector machines (RFESVMs) for linear or RELIEFF for non-linear separability (Schneider et al., 2017). In this way, the signal processing software as part of the sensor kit is realized in a highly modular design since heterogeneous sensors differ significantly regarding signal shape, time and spatial resolution, and target information to be extracted.

An example of the application of this toolbox is shown in Figs. 9 and 10. A miniaturized sensor system prototype was designed for integration in an electromechanical cylinder (EMC). These are increasingly applied as feed drives in machine tools, due to their unique combination of high loads, precision, and flexibility. The sensor system contains a range of (partially redundant) sensors (linear and rotary encoders, 3-D accelerometers, microphone, temperature and IR radiation sensors). Currently, the sensor prototype consists of two separate subsystems: first, two stacked sensor printed circuit boards (PCBs) (Fig. 9) mounted on the front surface of the ball screw inside the EMC housing (Festo ESBF-BS-63-400-5P, ∅ 63 mm, 400 mm stroke, 5 mm spindle pitch, axial load max. 7 kN) containing in total nine MEMS sensors. Furthermore, the rotary position of the spindle shaft is measured by an AMR Wheatstone bridge sensor (Doms and Slatter, 2014) with external bias magnet generating the support field which interacts with ferromagnetic teeth of the spindle shaft. This sensor is positioned at a fixed position in the cylinder housing close to the ball bearing (cf. Fig. 1a) pointing to the thread with a working distance of 1 mm. During rotation, the relative position of sensor and teeth changes, periodically resulting in sine and cosine sensor signals.

To evaluate the sensor system in a condition monitoring scenario, we induced a local abrasion of the spindle at stroke position 185 mm and recorded several stroke movements with varying velocity and three repetitions. For signal processing, short-time Fourier transform (STFT) was applied (length 10 000/overlap 2000 samples) with subsequent feature extraction and selection as previously demonstrated (Helwig et al., 2015b). Feature extraction captures a total of 210 statistical parameters such as median, variance, skewness, and kurtosis in different intervals of the amplitude spectra of three acceleration axes. The features are selected by F-value ranking of univariate ANOVA and dimensionally reduced to three discriminant functions (DFs) using LDA to obtain the maximum class separation. The latter algorithms are supervised learning methods, i.e. require class-annotated data which were given as velocity information and a local spindle condition traversed by the spindle nut. Figure 10a shows the resulting 3-D projection of sensor data with the planes DF1–DF2 and DF1–DF3 separating the different velocity levels and spindle conditions, respectively. Here, the velocity classes with 10, 20, and 50 mm s^{-1}, respectively, were used for training and the class with 30 mm s^{-1} velocity was used for evaluation. The intermediate velocity class fits well into the data-based model and the fault identification

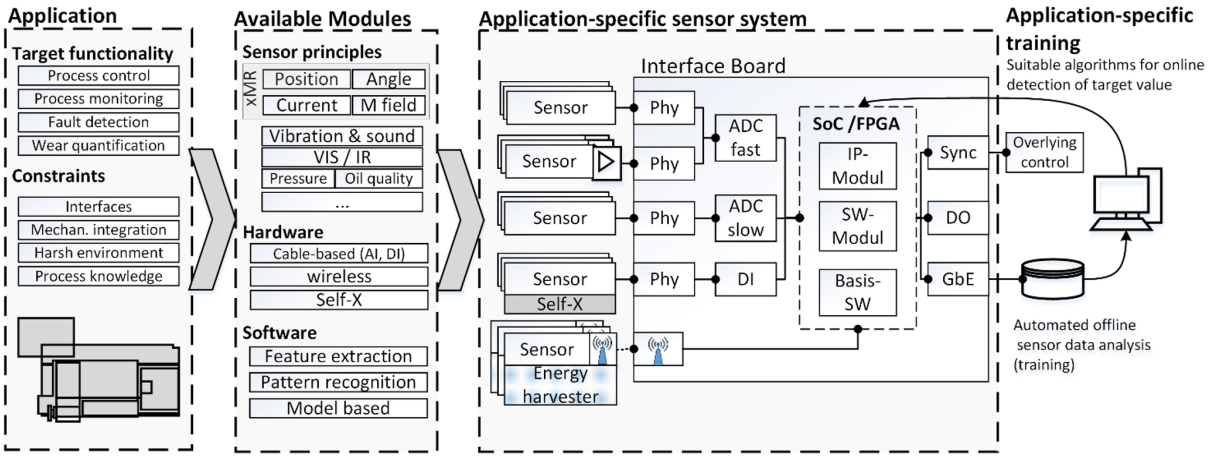


Figure 7. Application-specific design of the modular MoSeS-Pro sensor system kit combining various (micro-)sensors, especially xMR sensors, but also MEMS vibration, sound, pressure or IR radiation sensors, with electronics for data acquisition and pre-processing as well as communication interfaces in application-specific sensor systems (Helwig et al., 2017a).

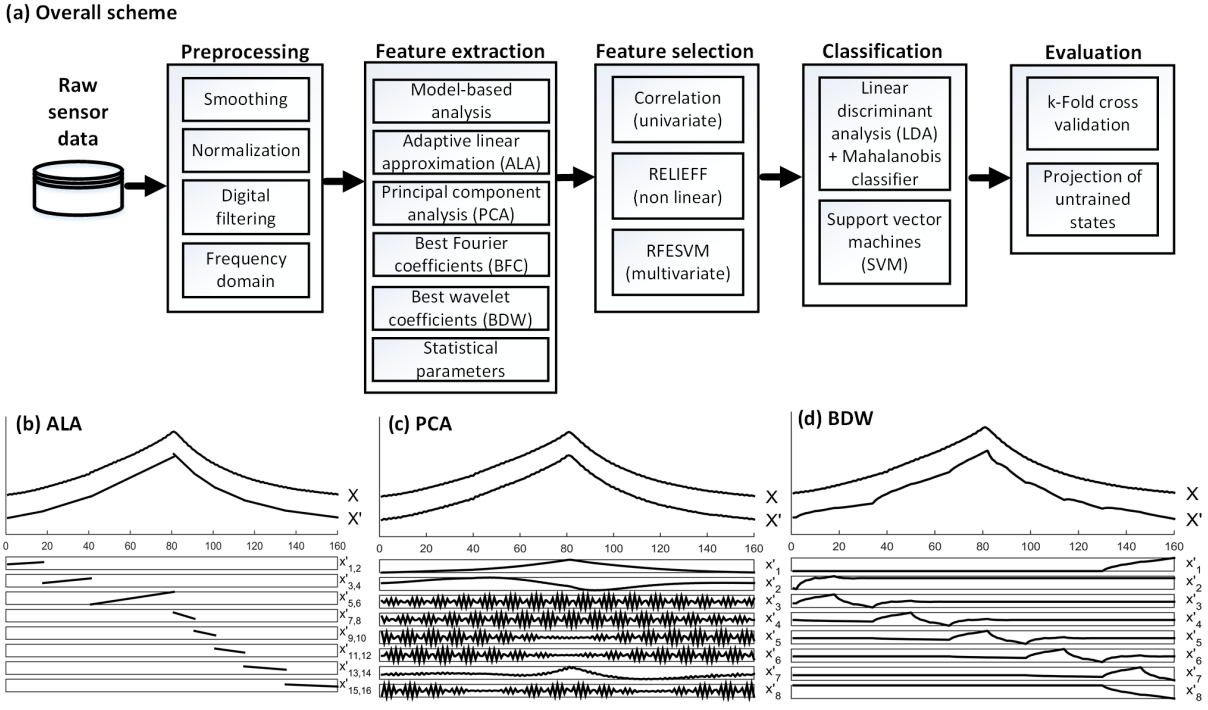


Figure 8. **(a)** Typical steps for offline data analysis and approximation of an exemplary sensor signal with different complimentary feature extraction methods: **(b)** adaptive linear approximation, **(c)** principal component analysis and **(d)** best wavelet coefficients using the largest Daubechies-4 wavelet coefficients (BDW: best Daubechies wavelet); in each case, X shows the original signal and X' the approximated signal using 16 **(b)** and 8 **(c, d)** features, respectively.

rate improves with increasing velocity. Figure 10b shows the plot of DF3 over stroke position clearly indicating the defect. The maximum is blurred, first, due to the interaction of balls and spindle defect over a distance of 30 mm and, second, also results from the STFT temporal blur. Furthermore, especially at low speeds with accordingly higher local resolution, two local maxima can be seen indicating the entry and exit points of the spindle nut passing over the defect. This example shows that the stroke position dependent analysis of signals can be used for fault diagnosis differentiating between local anomalies such as defects of the spindle and

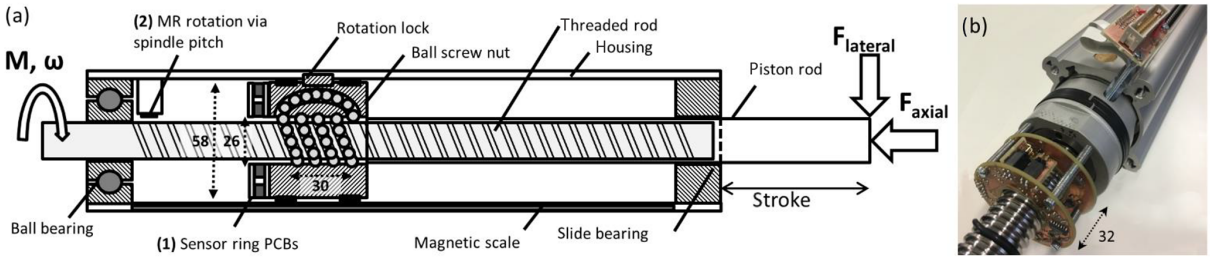


Figure 9. **(a)** Mechanical integration of the sensor system inside the EMC and **(b)** realization of the stacked sensor ring PCBs with the spindle shaft inside a disassembled cylinder (Helwig et al., 2017b).

global disturbances, e.g. of the ambient. For further details, the reader is referred to Helwig et al. (2017b).

To make full use of the MoSeS-Pro approach, data preprocessing and feature extraction need to be integrated into the sensor system to reduce the data load in the network and the cloud. However, this modular approach can also be used to design cost-efficient sensor systems for smart monitoring applications. In this case, a complete "over-instrumented" sensor set is used and the full sensor data are evaluated with the automated approach described above. Using the fairly simple and transparent algorithms allows identification of relevant sensors and features and, thus, the necessary acquisition bandwidth using an offline analysis. On this basis a greatly simplified sensor system can be defined for practical application. This approach would also allow us to choose an application-specific balance between sensor redundancy, i.e. to achieve robust operation as shown in Sect. 3, and cost efficiency.

5 Conclusion and outlook

Sensors and instrumentation are central driving forces for innovation, not only for Industry 4.0, but also for other megatrends that are described with the adjective smart, e.g. smart factory, smart production, smart mobility, smart home, or smart city. Intelligent decisions of complex systems are based on the knowledge of the system as well as ambient conditions and influence factors provided with high accuracy by sensors. The importance of sensors, measurement science, and smart evaluation for Industry 4.0 has been recognized and acknowledged by various authors (Imkamp et al., 2016; Sommer and Schütze, 2016; Walter, 2017) and has already led to the statement "Industry 4.0: nothing goes without sensor systems" ("*Industrie 4.0: Ohne Sensorsysteme geht nichts*") (Arnold, 2014). It should be acknowledged that notwithstanding all the euphoria and expectations for higher sensor production and sales volumes – especially when thinking about the Trillion sensor roadmap (Bryzek, 2013) – paradigm changes are expected, as is often the case in the digital revolution. Completely new business models like Uber and AirBnB already also exist in some sensor applications. Today, Google already provides the best traffic data based on mobile phone data with much better actuality and precision than classic traffic monitoring based on dedicated sensors. In this application the network plays an important role and of course the amount of data: while individual movement data provide low quality, data fusion of a large number of movements provides the required information. Similar effects can in the future also be expected for environmental data, i.e. air quality, when gas sensors are integrated into smartphones in large numbers. The field of sensors and measurement science and especially the research community have to address this challenge to ensure that future standards are still set by GMA (*VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik*), DKE (*Deutsche Kommission Elektrotechnik Elektronik Informationstechnik*), and AMA (in Germany), as well as BIPM, CEN/CENELEC and ISO worldwide, and not in Silicon Valley.

A possible approach for the sensor and measurement science community to play a bigger role in this development of Industry 4.0 might be the area of measurement uncertainty, which is simply not addressed by the computer science community today. In addition to making use of quantum standards integrated in smart sensors, an expanded view of the Guide for Expression of Uncertainty in Measurement (GUM, 2008) taking into account sensor data fusion and statistical modelling is highly desirable to make full use of the undisputed potentials and to continue with the success story of industrial production in high cost countries, which is one of the promises of Industry 4.0. Condition monitoring of complex production systems – from a single hydraulic press to a complete factory with assembly and test systems – can be one paradigm for the development of sensors and measurement science for Industry 4.0 as this immediately offers many economic advantages but can also be used for developing and testing new business models. A highly important aspect here is data security and with it the question of who owns which data and who has a right to access certain data. Consider a critical component being monitored in a complex production process: while the raw data are produced in the factory, the know-how for their interpretation lies with the component manufacturer. Forwarding complete process raw data to the

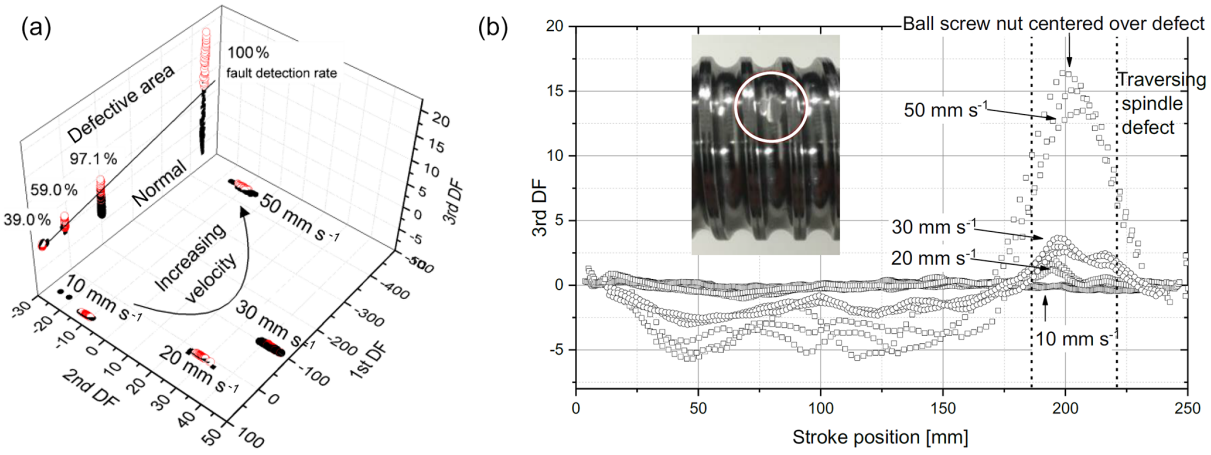


Figure 10. **(a)** LDA projection of 30 selected vibration features, $n = 2883$, with training based on velocity classes 10, 20, and 50 mm s^{-1}; classification rate determined for the Mahalanobis distance classifier with 10-fold cross-validation. **(b)** Deliberate abrasion as a local defect on the spindle and corresponding signal of DF3 vs. stroke position (moving average over 10 data points) (Helwig et al., 2017b).

component manufacturer is usually not an option, as this will also include confidential data, e.g. the production volume, from the factory. A successful business model will therefore require a certain level of trust between the involved partners but also a suitable abstraction level of the data from the component which would allow the required condition monitoring but no further insights into the confidential production process. This will of course be even more complicated if several component and sensor providers are involved to achieve the holistic condition monitoring approach based on data fusion. Perhaps this will lead to a new approach in data warehouse management with a novel type of neutral smart service provider to perform data anonymization and/or data analysis for all involved parties.

Data availability. The data are available in the UCI machine learning repository: https://archive.ics.uci.edu/ml/datasets/Condition+monitoring+of+hydraulic+systems.

Author contributions. NH was responsible for the iCM-Hydraulics project and performed the measurements and evaluations presented in Sect. 3. NH (hardware) and TS (software) were jointly responsible for the modular condition monitoring (CM) toolbox developed within the MoSeS-Pro project and the results presented in Sect. 4. AS coordinated the research within both projects and wrote the manuscript.

Competing interests. Andreas Schütze is a member of the editorial board of the journal.

Special issue statement. This article is part of the special issue "Evaluating measurement data and uncertainty". It is not associated with a conference.

Acknowledgements. The iCM-Hydraulics project received funding through the EFI programme (support of development, research, and innovation in Saarland), and the research by ZeMA was financed by HYDAC Filter Systems GmbH (Sulzbach, Germany).

The MoSeS-Pro project is sponsored by the German Federal Ministry of Education and Research in the call Sensor-based electronic systems for applications for Industrie 4.0 – SElekt I 4.0, funding code 16ES0419K, within the framework of the German Hightech Strategy.

Edited by: Klaus-Dieter Sommer
Reviewed by: three anonymous referees

References

Akmal Johar, M. and König, A.: Case Study of an Intelligent AMR Sensor System with Self-x Properties, in: Soft Computing in Industrial Applications, edited by: Gaspar-Cunha, A., Takahashi, R., Schäfer, G., and Costa, L., Springer, Berlin Heidelberg, 337–346, https://doi.org/10.1007/978-3-642-20505-7_30, 2011.

AMA Association for Sensors and Measurement (AMA Verband für Sensorik und Messtechnik e.V., originally Arbeitsgemeinschaft Messwertaufnehmer), available at: http://ama-sensorik.de/en, last access: 10 March 2018.

Arnold, H.: Kommentar Industrie 4.0: Ohne Sensorsysteme geht nichts, available at: http://www.elektroniknet.de/messen-testen/sonstiges/artikel/110776/ (last access: 10 March 2018), 2014.

Bastuck, M., Schütze, A., and Sauerwald, T.: A new approach to self-monitoring of amperometric oxygen sensors, Sensors and Actuators B 214, 218–224, https://doi.org/10.1016/j.snb.2015.02.116, 2015.

Baur, T., Schütze, A., and Sauerwald, T.: Optimierung des temperaturzyklischen Betriebs von Halbleitergassensoren, Tech. Mess., 82, 187–195, https://doi.org/10.1515/teme-2014-0007, 2015.

Bryzek, J.: Roadmap for the Trillion Sensor Universe, iNEMI Spring Member Meeting and Webinar, Berkeley, CA, 2 April, available at: http://www-bsac.eecs.berkeley.edu/scripts/show_pdf_publication.php?pdfID=1365520205 (last access: 10 March 2018), 2013.

Cachay, J. and Abele, E.: Developing Competencies for Continuous Improvement Processes on the Shop Floor through Learning Factories – Conceptual Design and Empirical Validation, Procedia CIRP, 3, 638–643, https://doi.org/10.1016/j.procir.2012.07.109, 2012.

Digitale Transformation in der Industrie, available at: http://bmwi.de/DE/Themen/Industrie/industrie-4-0.html, last access: 10 March 2018.

Doms, M. and Slatter, R.: Magnetoresistive sensors for angle, position, and electrical current measurement in demanding environments, Proc. SPIE 2014, 9113, https://doi.org/10.1117/12.2049886, 2014.

Duda, R. O., Hart, P. E., and Stork, D. G.: Pattern classification, 2 Edn. Wiley, New-York, 2000.

Gassmann, O. and Kottmann, J.: Technologiemanagement in der Sensorik, Wissensmanagement, 8, 19–24, 2002.

GUM: Evaluation of measurement data – Guide to the expression of uncertainty in measurement, JCGM 100, available at: http://www.bipm.org/en/publications/guides/gum.html (last access: 10 March 2018), 2008.

Helwig, N. and Schütze, A.: Intelligentes Condition Monitoring mit automatisierter Merkmalsgenerierung und -bewertung, in: XXVIII. Messtechnisches Symposium des Arbeitskreises der Hochschullehrer für Messtechnik, edited by: Schütze, A. and Schmitt, B., Shaker Verlag, Aachen, 121–128, https://doi.org/10.5162/AHMT2014/P1, 2014.

Helwig, N. and Schütze, A.: Detecting and compensating sensor faults in a hydraulic condition monitoring system. Proc. SENSOR 2015 – 17th International Conference on Sensors and Measurement Technology, Nuremberg, 19–21 May, available at: https://doi.org/10.5162/sensor2015/D8.1, 2015.

Helwig, N., Pignanelli, E., and Schütze, A.: Condition Monitoring of a Complex Hydraulic System Using Multivariate Statistics, Proc. I2MTC-2015 – 2015 IEEE International Instrumentation and Measurement Technology Conference, paper PPS1-39, Pisa, Italy, 11–14 May, available at: https://doi.org/10.1109/I2MTC.2015.7151267, 2015a.

Helwig, N., Klein, S., and Schütze, A.: Identification and quantification of hydraulic system faults based on multivariate statistics using spectral vibration features, Proc. Eng., 120, 1225–1228, https://doi.org/10.1016/j.proeng.2015.08.835, 2015b.

Helwig, N., Schneider, T., and Schütze, A.: MoSeS-Pro: Modular sensor systems for real time process control and smart condition monitoring using XMR-technology, Proc. 14th xMR-Symposium "Magnetoresistive Sensors and Magnetic Systems", Wetzlar, Germany, 21–22 March, 2017a.

Helwig, N., Merten, P., Schneider, T., and Schütze, A.: Integrated Sensor System for Condition Monitoring of Electromechanical Cylinders, MDPI Proceedings 2017, 1, 626, https://doi.org/10.3390/proceedings1040626, 2017b.

Hightech-Strategie: Industrie 4.0; available at: http://www.hightech-strategie.de/de/Industrie-4-0-59.php, last access: 10 March 2018.

iCM Hydraulics – Data-based intelligent condition monitoring for hydraulic systems; project funded in the EFI program of Saarland, subcontract by HYDAC Filter Systems GmbH, performed at Centre for Mechatronics and Automation gGmbH (ZeMA), 2013–2015.

Imkamp, D., Berthold, J., Heizmann, M., Kniel, K., Manske, E., Peterek, M., Schmitt, R., Seidler, J., and Sommer, K.-D.: Challenges and trends in manufacturing measurement technology – the "Industrie 4.0" concept, J. Sens. Sens. Syst., 5, 325–335, https://doi.org/10.5194/jsss-5-325-2016, 2016.

Industrial Internet Consortium, available at: http://www.iiconsortium.org, last access: 10 March 2018.

Kitching, J., Donley, E. A., Knappe, S., Hummon, M., Dellis, A. T., Sherman, J., Srinivasan, K., Aksyuk, V. A., Li, Q., Westly, D., Roxworthy, B., and Lal, A.: NIST on a Chip: Realizing SI units with microfabricated alkali vapour cells, Journal of Physics: Conference Series, 723, 012056, https://doi.org/10.1088/1742-6596/723/1/012056, 2016.

MoSeS-Pro: Modulare Sensorsysteme für Echtzeit-Prozesssteuerung und smarte Zustandsbewertung für die Industrie 4.0, BMBF project funded in the funding area "Sensorbasierte Elektroniksysteme für Anwendungen für Industrie 4.0 (SElekt I4.0)", available at: http://www.moses-pro.de/ (last access: 10 March 2018), 2015–2018.

Marien, J. and Schütze, A.: Magnetic Microsensors: Quo vadis?, Proc. SENSOR 2009, II, 17–22, Nuremberg, 26–28 May, available at: https://doi.org/10.5162/sensor09/v2/a6.1, 2009.

Munter, P. J. A.: A low-offset spinning-current hall plate, Sensors and Actuators B, 22, 743–746, https://doi.org/10.1016/0924-4247(89)80069-X, 1990.

NIST-on-a-Chip Portal, available at: http://www.nist.gov/pml/productsservices/nist-chip-portal, last access: 10 March 2018.

Ochs, T.: Selbstüberwachung und online Verifizierung von Sensordaten im Kraftfahrzeug, Übersichtsvortrag, 11. Dresdner Sensorsymposium 2013, Dresden, 9–11 December 2013.

Plattform Industrie 4.0, available at: http://www.plattform-i40.de/I40/Navigation/EN/Home/home.html, last access: 10 March 2018.

Reimann, P. and Schütze, A.: Sensor Arrays, Virtual Multisensors, Data Fusion, and Gas Sensor Data Evaluation, in: Gas Sensing Fundamentals, edited by: Kohl, C.-D. and Wagner, T., Springer Series on Chemical Sensors and Biosensors, Volume 15, 2014.

Schneider, T., Helwig, N., and Schütze, A.: Automatic feature extraction and selection for classification of cyclical time series data, Tech. Mess., 84, 198–206, https://doi.org/10.1515/teme-2016-0072, 2017

Schüler, M., Sauerwald, T., and Schütze, A.: A novel approach for detecting HMDSO poisoning of metal oxide gas sensors and improving their stability by temperature cycled operation, J. Sens. Sens. Syst., 4, 305–311, https://doi.org/10.5194/jsss-4-305-2015, 2015.

Schütze, A.: Sensorik und Messtechnik im Industrie 4.0-Zeitalter, Plenarvortrag, 7. VDI-Fachtagung Messunsicherheit 2015 – Messunsicherheit praxisgerecht bestimmen, Braunschweig, Germany, 19–20 November, 2015.

Schütze, A. and Helwig, N.: Sensorik und Messtechnik für die Industrie 4.0 – (Sensors, instrumentation and measurement science for "Industrie 4.0"), Tech. Mess., 83, 208–218, https://doi.org/10.1515/teme-2016-0047, 2016.

Schütze, A. and Helwig, N.: Sensors 4.0 – Smart sensors and measurement technology enable Industry 4.0, Proc. 14th xMR-Symposium "Magnetoresistive Sensors and Magnetic Systems", Wetzlar, Germany, 21–22 March, 2–8, 2017.

Semantic Sensor Network Ontology, available at: http://www.w3.org/2005/Incubator/ssn/ssnx/ssn, last access: 10 March 2018.

Sensor-Trends 2014 – Trends in zukunftsorientierten Sensortechnologien, edited by: AMA Fachverband für Sensorik, available at: http://www.ama-sensorik.de/fileadmin/Pubikationen/AMA_Trendbericht_Langfassung[1].pdf (last access: 10 March 2018), 2010.

Sommer, K.-D. and Schütze, A.: Smart sensors & networked digital measurement systems – Trends and challenges in industrial measurement and metrology, Keynote lecture, 46th Ann. Meas. Science Conf. 2016, Anaheim, USA, 23–25 March, 2016.

Stahl-Offergeld, M.: Robuste dreidimensionale Hall-Sensoren für mehrachsige Positionsmesssysteme, "Aktuelle Berichte aus der Mikrosystemtechnik – Recent Developments in MEMS", Band 20, Shaker-Verlag, Aachen, 2011.

Stahl-Offergeld, M., Cichon, D., Hohe, H., and Schütze, A.: Offset Tracing in Hall Sensors by Integrated Temperature Coefficient Determination, Proc. SENSOR 2009, II, 59–64, https://doi.org/10.5162/sensor09/v2/a7.4, 2009.

Stahl-Offergeld, M., Ernst, R., Hohe, H.-P., and Schütze, A.: Process-independent Integrated Sensitivity Calibration of 3D Hall Sensors, EMSA 2010, the 8th European Conference on Magnetic Sensors and Actuators, Bodrum, Turkey, 4–7 July, 2010.

Walter, K.-D.: Wo bleibt der Sensor für Industrie 4.0?, available at: http://www.elektrotechnik.vogel.de/wo-bleibt-der-sensor-fuer-industrie-40-a-529141/, last access: 10 March 2018.

5.2 Paper 1: Automatic Feature Extraction and Selection for Classification of Cyclical Time Series Data

Given the economic and technical restrictions imposed on smart sensors, a set of mutually complementing algorithms is proposed and introduced. The individual algorithms and their suggested combinations are shown and demonstrated in four different examples, namely:

- Process sensors of a hydraulic test bed to detect accumulator pressure loss,
- Vibration sensors on a hydraulic test bed to detect accumulator pressure loss and degradation of cooling power,
- Gas sensors with temperature cycled operation (TCO) to quantify Naphthalene concentration,
- Impedance spectroscopy of a gas sensor to discriminate different gases and detect sensor poisoning.

Additionally, a random dataset was generated based on the gas sensor with TCO to demonstrate the absence of overfitting in algorithm selection. As intended, the automated machine learning toolbox successfully failed to surpass random guessing on the randomized dataset while achieving better performance in comparison with algorithms previously applied on the other datasets.

Automatic Feature Extraction and Selection for Classification of Cyclical Time Series Data

Tizian Schneider[1], Nikolai Helwig[1], and Andreas Schütze[1,2]

[1]Saarland University, Lab for Measurement Technology, Saarbrücken, Germany
[2]Centre for Mechatronics and Automation Technology (ZeMA gGmbH), Saarbruecken, Germany

tm – Technisches Messen (2016), 84 (3), 198-206

The original paper can be found online at https://doi.org/10.1515/teme-2016-0072.

© Used with permission of Walter de Gruyter and Company, from Automatic Feature Extraction and Selection for Classification of Cyclical Time Series Data, Schneider, Tizian; Helwig, Nikolai; Schütze, Andreas, 84, 3, 2018; Permission was conveyed through Copyright Clearance Center, Inc.

Tizian Schneider*, Nikolai Helwig, and Andreas Schütze

Automatic feature extraction and selection for classification of cyclical time series data

Automatische Merkmalextraktion und -selektion von Merkmalen zur Klassifikation zyklischer Signalverläufe

DOI 10.1515/teme-2016-0072
Received November 30, 2016; revised January 16, 2017; accepted January 16, 2017

Abstract: The classification of cyclically recorded time series plays an important role in measurement technologies. Example use cases range from gas sensors combined with temperature cycled operation to condition monitoring using vibration analysis. Before machine learning can be applied to high dimensional cyclical time series data dimensionality reduction has to be performed to avoid the classifier suffering from overfitting and the "curse of dimensionality". This paper introduces a set of four complementary feature extraction methods and three feature selection algorithms that can be applied in a fully automatized manner to reduce the number of dimensions. The feature extraction algorithms are capable of extracting characteristic features from cyclical time series catching information contained in local details and overall cycle shape as well as in frequency or time-frequency domain. The methods for feature selection are capable of selecting the most suitable features for linear and nonlinear classification. The methods were chosen to be applicable to a wide range of applications which is verified by testing the set of methods on four different use cases.

Keywords: Time series classification, dimensionality reduction, machine learning.

Zusammenfassung: Die Klassifikation zyklischer Signalverläufe mittels maschinellen Lernens spielt eine wichtige Rolle in der Messtechnik. Beispielanwendungen reichen von Gas-Sensoren, die temperaturzyklisch betrieben werden, bis hin zur Zustandsüberwachung durch Vibrationsanalyse. Bevor maschinelles Lernen auf die hochdimensionalen, zyklischen Signalverläufe angewandt werden kann, muss deren Dimensionalität verringert werden, um zu verhindern, dass der Klassifikator unter Overfitting und dem „curse of dimensionality" leidet. In dieser Veröffentlichung werden vier sich gegenseitig ergänzende Methoden zur Merkmalextraktion und drei Algorithmen zur Merkmalselektion vorgeschlagen, die automatisiert genutzt werden können, um die Dimensionalität zu verringern. Die Algorithmen zur Merkmalextraktion extrahieren charakteristische Merkmale aus dem Signalverlauf. Die in den Merkmalen enthaltene Information beinhaltet dabei nicht nur lokale Details und die allgemeine Kurvenform sondern auch Merkmale aus dem Frequenz- und Zeit-Frequenz-Bereich. Die Methoden zur Merkmalselektion sind in der Lage die besten Merkmale für lineare und radiale Klassifikation auszuwählen. Die Methoden wurden so ausgewählt, dass sie für ein möglichst breites Anwendungsspektrum geeignet sind, was durch die erfolgreiche Anwendung auf vier verschiedene Beispieldatensätze gezeigt wird.

Schlüsselwörter: Klassifikation von Signalverläufen, Dimensionsreduktion, maschinelles Lernen.

***Corresponding author: Tizian Schneider,** Saarland University, Dept. of Mechatronics Engineering, Lab for Measurement Technology, P.O. Box 15 11 50, 66041 Saarbruecken, Germany; and ZeMA – Center for Mechatronics and Automation Technology gGmbH, Eschberger Weg 46, Buisness Park, Building 9, 66121 Saarbrücken, Germany, e-mail: t.schneider@zema.de
Nikolai Helwig, Andreas Schütze: Saarland University, Dept. of Mechatronics Engineering, Lab for Measurement Technology, P.O. Box 15 11 50, 66041 Saarbruecken, Germany; and ZeMA – Center for Mechatronics and Automation Technology gGmbH, Eschberger Weg 46, Buisness Park, Building 9, 66121 Saarbrücken, Germany

1 Introduction

Time series play an important role in measurement technologies. Thereby cyclical time series, which are for example recorded during a fixed working cycle of an industrial machine, are of special interest. Classification of such signals can be used for condition monitoring of the machine. If machine faults have different effects during different parts of the machine working cycle the fault mechanism

produces a certain "fingerprint" in the cyclical sensor data that can be used to determine the fault progress allowing predictive maintenance [1, 2]. Using machine learning trained on a database containing cycles coming from machines with and without faults these fingerprints can be identified. The problem addressed in this paper is that common machine learning algorithms are not suitable for time series classification due to the high dimensionality of raw data, if each measurement point in a cycle is treated as a new dimension. One issue is the computational complexity of the learning algorithm making learning in high-dimensional space inefficient. More important, however, are effects like overfitting and the "curse of dimensionality" [3] which reduce classifier performance in high-dimensional space.

2 Dimensionality reduction

To apply machine learning to high-dimensional data, suitable dimensionality reduction has to be performed by extracting characteristic features from the time series (feature extraction) and selecting the most relevant features (features selection). Feature extraction is an unsupervised (i.e. is done without knowledge of the cycle's group affiliation) process that only depends on the basic shape of the cycle which should be represented as accurate as possible by the features. However, feature selection selects the most relevant features with respect to a given classification target. Therefore, feature selection is a supervised process which depends on knowing the correct group affiliation of each cycle.

To reduce the effort spent on dimensionality reduction, this paper suggests a complementary set of automated methods for feature extraction and selection. Since there is no universal method for dimensionality reduction of time series a set of algorithms is better suited to cover a wide range of possible applications as a single algorithm. The methods suggested for feature selection were chosen for working best on different applications of current research, i.e. by offering complementary performance. Additionally, the individual algorithms also have to be as effective and efficient as possible. Efficiency in the context of time series means being scalable, both with respect to a high number of data points per cycle, to account for long cycles and high sampling rates, and a high number of cycles, to account for large databases. Effectiveness of feature extraction is the ability of the algorithm to represent the original cycle as accurately as possible using as few relevant features as possible. In the process the algorithm must not create new clusters within the database because new clusters increase model complexity needed to learn classification rules for fault types. Creating new clusters is a typical risk when using feature extraction methods that use features for bookkeeping. With respect to the consecutive feature selection the feature extraction algorithm should concentrate the relevant information in few features to allow feature selection to easily identify and select the most relevant features.

3 Feature extraction

Each of the four feature extraction algorithms described in this chapter is suitable for representing different types of information typically contained in time series data. In this sense the algorithms complement each other to process data from a broad spectrum of real world applications.

The four methods suggested for feature extraction after comparison of a total of 15 methods [4] are:

3.1 Adaptive linear approximation (ALA)

Information contained in local details like transients and edges can be represented by piecewise approximation of the cycle with linear functions. Piecewise approximation splits the cycle into several variable length segments which are represented by the linear fit parameters mean value and slope. This is particularly useful if relevant information is only contained in certain cycle segments because relevant information is concentrated in few features allowing feature selection to safely discard irrelevant information. The linear fit function is chosen because it is computationally cheap and provides first order approximations. Since irrelevant signal components like noise require higher order approximations they are efficiently suppressed. In addition, multiple automated algorithms are available for linear segmentation. The only algorithm which guarantees minimal regression error over all cycles for a given number of segments is described in [5]. This algorithm also automatically suggests a reasonable number of segments to use and, thus, automatically tunes the tradeoff between low approximation error and low number of features. Figure 1(l) shows the resulting approximation of the a gas sensor response to TCO.

3.2 Principal Component Analysis (PCA)

The best way to catch the general trend of a cycle in the time domain is to use the projection onto the first principal

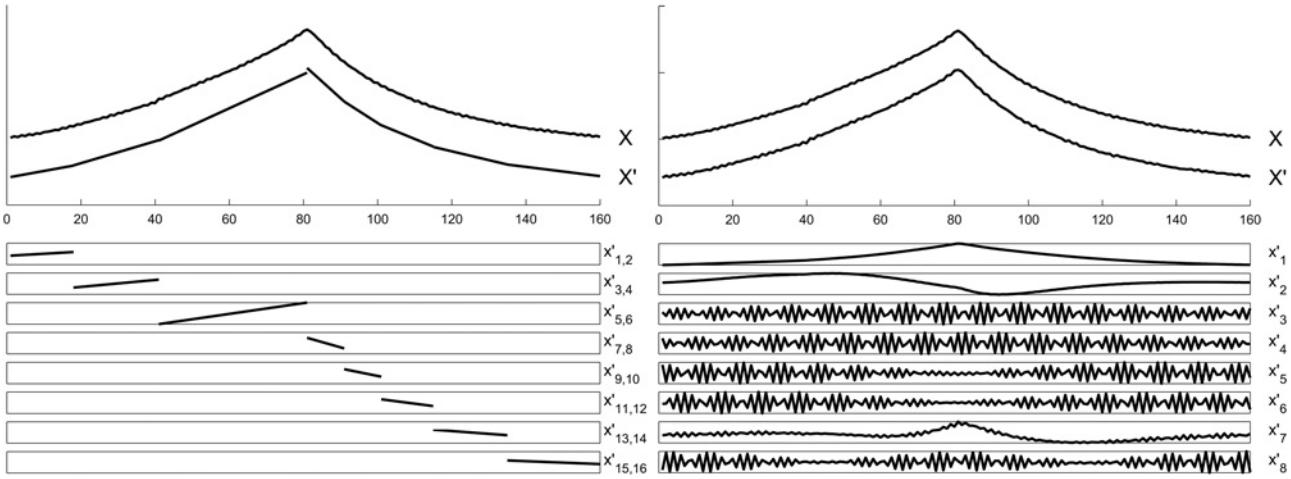


Figure 1: (l) Approximation X' using ALA with features x'_1–x'_{16}, compared to the original gas sensor temperature cycle response X (shifted for better clarity). The eight segments are represented by mean values (uneven indices) and slopes (even indices). (r) Approximation using PCA. Extracted features x'_1–x'_8 are projections onto the first principal components that are shown below.

components (PCs) found by PCA. This method is chosen because the first principal components are the best linear transformation of the dataset for a given number of features (projections on PCs) in terms of approximation error [6]. An alternative view to PCA is that the first PCs are formed to explain maximum variance and therefore capture fundamental cycle trends while later PCs capture details and noise. This enables the PCA to determine the number of PCs required to represent all non-random variation in the training dataset using Bartlett's test [7] for equal variance. Since the last PCs capturing random noise have equal variance they can be safely discarded. Because PCA is a global transformation of the dataset trying to minimize the global approximation error, local details are rarely represented by the first principal components and are therefore often neglected. Thus, PCA is good for a representation of overall trends and decomposition of the signal into different characteristics influenced by different driving forces, while ALA is good to represent localized information. How the decomposition of the signal into different characteristics can look like is shown in Figure 1(r). It is clearly visible, how the signal is a consists of an actual sensor response (visible in principal components one, two, and seven) and a superimposed beat, that is induced by sensor electronics and does not carry any relevant information for gas sensing.

3.3 Best Fourier Coefficients (BFC)

Depending on the nature of the measurement variable the relevant information might not be represented well in the time domain. Especially in vibration analysis the information is expected to be much better represented in the frequency domain. To capture this information Fourier transformation is applied to transform raw data into the frequency domain. Extracted features are amplitudes and phase shifts. To reduce the dimensionality initially the symmetry property of the Fourier transformation is exploited which allows discarding half of the Fourier coefficients without loss of information. Then, only the ten percent of the coefficients with highest mean absolute value over all cycles are extracted as features. As described in [8], choosing these coefficients will preserve maximum signal energy for a reduction factor of ten. Note that using another reduction factor than ten enables the user to control the tradeoff between low approximation error and low number of extracted features. Figure 2(l) shows the resulting approximation when using BFC. As seen the overall cycle shape is more dominant than the superimposed beat (compare Figure 1(r)) and the beat is therefore discarded. The same accounts for the sharp edge at measurement point 80, that is represented by multiple high frequency coefficients with low amplitude.

3.4 Best Daubechies-Wavelet Coefficients (BDW)

Wavelet decomposition is applied to capture information in both the time and frequency domains. The Daubechies-4 Wavelet has been chosen because of its widespread use in signal processing and data compression. Wavelet decomposition is particularly interesting for providing

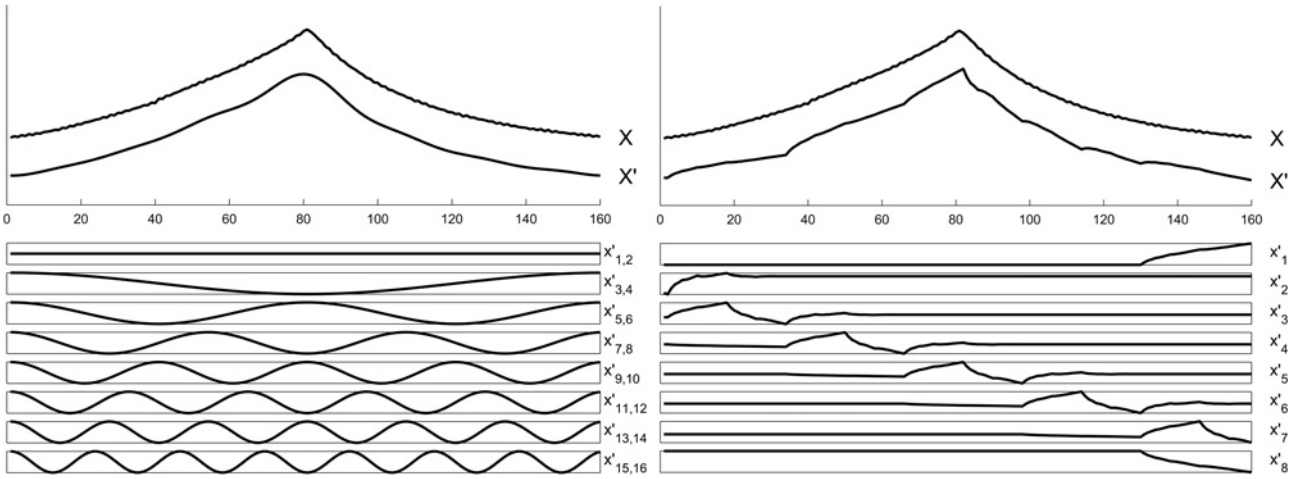


Figure 2: (l) Approximation X' using BFC with features x'_1–x'_{16} compared to the original gas sensor temperature cycle response X (shifted for better clarity). The approximation is the superposition of multiple sinus waves represented by amplitude (features with uneven indices) and phase shifts (features with even indices). (r) Approximation using BDW. Extracted features are wavelet coefficients x'_1–x'_8.

a multiresolution view of the data since the first wavelet coefficients are extracted from global signal sections while the last coefficients are extracted from local details [8]. As for Fourier transformation the coefficients with highest mean absolute value over all cycles are extracted as features. Again, this method preserves the maximum signal energy for a given reduction factor and the reduction factor can be used to control the tradeoff between accuracy and number of features. The approximation using BDW is shown in Figure 2(l).

4 Feature selection

After feature extraction feature selection is applied to select the most relevant features for machine learning. As Houle et al. suggest [9] the "curse of dimensionality" is not caused by high dimensionality per se but by many features providing low contrast regarding classification. Thus, choosing only the most relevant features and discarding low contrast features efficiently suppresses the curse of dimensionality. Again, a huge variety of algorithms is available; however, none of these guarantees to find the optimal solution for all types of data. Therefore, the following, complementary set of algorithms is suggested:

4.1 Recursive Feature Elimination Support Vector Machines (RFESVM)

RFESVM is an embedded method for feature selection combining machine learning based on Support Vector Machines (SVM) with feature selection. The algorithm recursively trains a linear SVM on the current feature set and eliminates the feature with least relevance for group separation [10, 11]. As SVMs only address binary classification multiclass problems are resolved in the implementation using one vs. one multiclass encoding and averaging feature relevance over all binary classifiers. In addition, the parameter C of the binary SVMs, which controls the trade-off between training error and margin maximization (with $C = \infty$ leading to a hard margin SVM) [12], was set to a fixed value of 1,000 and all features are standardized before each SVM training. In addition, RFESVM was only applied to the 500 features with highest Pearson correlation to the target value to limit the computational effort.

4.2 Pearson correlation

Pearson correlation is used for quick results and feature pre-selection due to its low computational cost. Pearson correlation ranks features according to their individual linear correlation to the target. Both RFESVM and Pearson correlation select features for linear classification, but in contrast to Pearson correlation RFESVM takes into account feature interaction and any kind of monotonic relationship between features and target. This enables RFESVM to find more effective feature subsets and to eliminate redundancies that will occur when selecting the highest correlated features.

4.3 Univariate RELIEFF

For classes that are not linearly separable it seems natural to use learning algorithms based on nonlinear classification. RELIEFF is based on radial classification using k Nearest Neighbors (kNN). The implementation used is the univariate variant of the algorithm described in [13] and computes a relevance index based on kNN with k = 5. Since RELIEFF is based on radial classification it is complementary to RFESVM which is based on linear classification. However, the implementation of RELIEFF in this paper rates the relevance of features individually, unlike RFESVM which rates features in the context of other features. RELIEFF is applied to the 1,000 highest correlated features.

For all methods the optimal number of features is estimated by computing the tenfold cross-validation error of a Linear Discriminant Analysis (LDA) in combination with Mahalanobis classification as described in [14]. This means the group affiliation of a new point is predicted by assigning them to the group with minimum Mahalanobis distance to the point. Since LDA is computationally cheap the cross-validation error is computed for all feature sets with less than 500 features to select the number of features that yields minimum classification error. Note that any other classification algorithm that is suitable for the learning problem at hand could be used for the final classification step. The rather simple LDA is chosen here to demonstrate that dimensionality reduction does not increase model complexity and keeps the learning problem simple when applied as suggested in this paper.

5 Application of feature extraction and selection

The algorithms are combined by evaluating datasets with all combinations of the named feature extraction and selection algorithms as shown in Figure 3. This combination effectively finds the best features for linear and radial classification in time, frequency and time-frequency domain. Please note that this method, like every other one, does not guarantee finding the optimal feature representation. Nevertheless, the combination of complementary methods provides a high probability to find a relevant feature subset. To prove this concept and to show the applicability for different types of data the suggested set of methods was successfully applied to the following datasets from four different applications and an additional synthetic data-set to verify the robustness of the algorithms against overfitting.

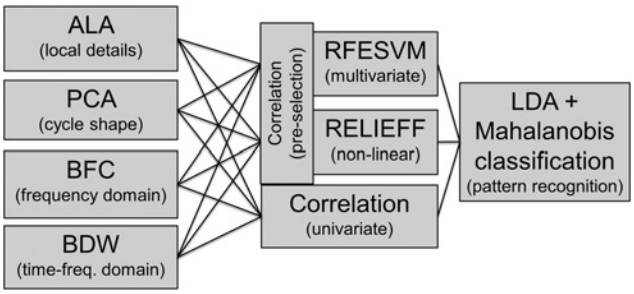


Figure 3: Schematic of the suggested algorithms for feature extraction (left), feature selection (middle) and classification (right) and their combination for automated dimensionality reduction and classification. Abbrevations: Adaptive Linear Approximation (ALA), Principal Component Analysis (PCA), Best Fourier Coefficients (BFC), Best Daubechies Wavelets (BDW), Recursive Feature Elimination Support Vector Machines (RFESVM), Linear Discriminant Analysis (LDA).

- Temperature cycled gas sensors [4, 14]
- Impedance spectroscopy of gas sensors [15]
- Condition monitoring using process sensors [1]
- Condition monitoring using vibration analysis [16]
- Random data

6 Results

Since this paper is focused on feature extraction and selection only results achieved using LDA are reported here. Since LDA is working best for classes forming single Gaussian distributed clusters low cross-validation errors reflect the success of representing data in an easily accessible way. All errors are reported as percentage of misclassified cycles based on tenfold cross-validation with LDA projection and consecutive classification using Mahalanobis distance to the class centers. Furthermore all the tested datasets have been evaluated using LDA in the respective paper, which makes it possible to trace all achieved results back to improved feature extraction and selection. Note that any other machine learning algorithm could have been used for classification.

6.1 Temperature cycled gas sensor

The first use case shown in this paper is the temperature cycled operation (TCO) of gas sensors. The goal here is to determine the concentration of hazardous naphthalene in the ppb range independent of varying ethanol background

as required for indoor air quality classification [14]. During training the gas sensor was exposed to six different concentrations of naphthalene, each in three different background concentrations of ethanol. A total of 1569 cycles with 160 data points, i.e. measurements of sensor conductance, per cycle were recorded [4]. The temperature cycled gas sensor can suffer from sensor drift and a varying concentration of background gas that strongly influences the sensor signal. These nonstationary effects prevent the application of time series modeling algorithms like Hidden Markov Models that require the signal to be stationary. Using the suggested approach allows the learning algorithm to take into account these effects and to compensate for them (with LDA only linearly).

For this dataset the choice of the feature extraction algorithm does not affect the classification performance significantly since all representations of the cycles lead to very low classification errors below 1%. However, feature selection is much more relevant to reduce the cross-validation error and reveal the basic structure of the data. This can be shown by comparing ALA applied with and without consecutive feature selection. When applied without feature selection the 20 features extracted by ALA lead to a cross-validation error of 5% while the best two features selected by RFESVM yield an error of 0.3%. This effect can be traced back to the fact, that the targeted naphthalene concentrations have been offered in three different background gases and do therefore not form Gaussian distributed clusters in feature space. Given the fact that LDA finds the low dimensional projection of the feature-space that maximizes the ratio of between class scattering and within class scattering, LDA uses the additional features in the full set to minimize the within class scattering by projecting the within group clusters, that are formed by varying background, onto another. This effect increases overlap between different groups and is prevented by feature selection since feature selection removes the feature that allow LDA to misfit the sample distribution. The corresponding two-dimensional LDA-projections can be found in Figure 4. Although this problem is caused by the classifier used and could be circumvented using a nonlinear classifier it shows how feature selection can simplify a learning problem which comes in handy when applied to a high dimensional problem that cannot be solved directly with a nonlinear classifier.

6.2 Impedance spectroscopy of gas sensor

In the second use case impedance spectroscopy was applied to a gas sensor to discriminate different gases and to simultaneously assess the sensor's degradation state that is increased by several stages throughout the measurement by sensor poisoning. The goal is to find a classifier that is capable of determining gas type and degradation state. A total of 7983 complex spectra containing measurements of admittance at 801 different frequencies were recorded [15]. For dimensionality reduction each spectrum is treated as one cycle consisting of two sensors measuring real and imaginary part of the admittance. Feature extraction is applied to each sensor separately. Afterwards features extracted by the same method are grouped

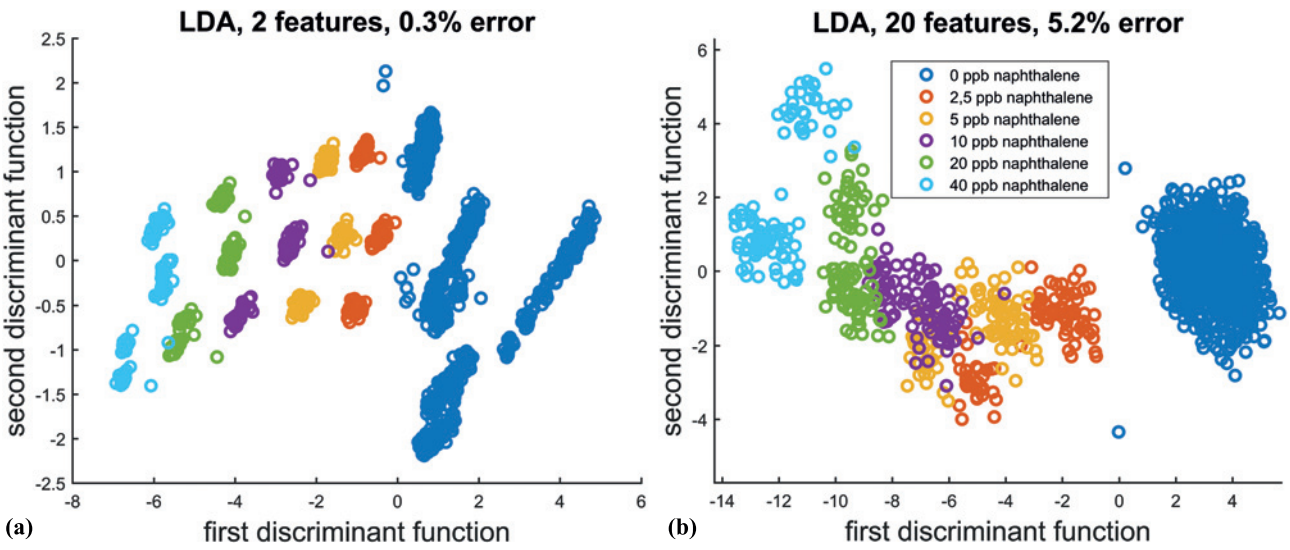


Figure 4: (a) LDA scatter plot of TCO gas sensor data using two most relevant features. (b) LDA scatter plot of TCO gas sensor using 20 features. Abbrevations: Linear Discriminant Analysis (LDA), parts per billion (ppb).

together before feature selection is applied. This approach is used for all discussed datasets that contain more than one sensor. For more details see [15].

For this problem the best classification results were achieved using a combination of ALA and RFESVM. The reported classification error is 1.8%. This result shows that the proposed set of methods can easily be applied to evaluate data coming from multiple sensors. Furthermore this shows the applicability to spectral data. It is worth noting, that the classification error increases to 4.6% when feature extraction is performed on sensor impedance instead of admittance. The fact, that impedance and admittance contain the exact same information but yield different classification errors, shows the importance of data representation for machine learning. This emphasizes the importance of feature extraction and selection, which can be further increased by improving the raw data by preprocessing.

6.3 Condition monitoring using process sensors

In the third use case machine learning is applied to classify the progression of different fault mechanisms of a complex hydraulic system. During training the hydraulic machine is operated with a fixed work cycle. The setup of the test system enables the operator to simulate different fault mechanisms like accumulator pressure decrease in different states of progression. During training 1,449 cycles with multiple simulated faults were recorded. A total of 15 process sensors for pressure, temperature, flow, electrical power and vibration were used to generate signals between 60 and 6,000 measurement points per cycle. The goal is to classify the progression of a single fault-mechanism independent of all other fault-mechanisms. For more details see [1].

In the previous work by Helwig et. al. [1] the working cycle of the hydraulic machine was split into thirteen segments that each cover a static or transient part of the cycle. Subsequently from all segments of all sensor signals the features median, variance, slope, position of maximum, skewness and kurtosis are extracted. From this feature set the 20 features with highest absolute Pearson correlation to the respective target are selected for LDA training. In comparison with this method the automated approach achieved improvements of the classification rate while simultaneously the manual effort for feature extraction was significantly reduced. Regarding classification performance the highest increase was achieved for accumulator pressure of the hydraulic system. The previously reported error rate of 9.6% [1] was decreased to only 0.35% using a combination of ALA and RFESVM. There was no significant improvement in classification performance for the other possible fault mechanisms since the previously achieved results were already very good preventing significant further improvement. Nevertheless, in these cases the effort required for feature extraction by manually defining the cycle segments from which features are extracted was greatly reduced by ALA automatically identifying linear segments. The good results achieved on this dataset show that the statistical approach is still applicable if the machine is too complex for physical modeling.

6.4 Condition monitoring using vibration analysis

The last use case is vibration analysis for condition monitoring of the hydraulic system described above. Instead of process sensors an accelerometer at the main oil pump was used to record vibrations during the 975 working cycles with 483,328 points per cycle. For more details see [16].

The application of the described set of features to vibrational data both shows the scalability of the methods to nearly half a million measurement points per cycle and the applicability for information best represented in frequency and time-frequency domains. Furthermore it shows that the best data representation depends on the respective training target, i.e. the fault mechanism to be quantified. For example the decrease in cooling power is better represented by raw time domain data (perfect classification using highest correlated Daubechies-4 Wavelet coefficients) than by the amplitude spectrum after Fourier transformation applied as preprocessing (25% error using highest correlated Daubechies-4 Wavelet coefficients). On the other hand, the decrease in accumulator pressure is better represented in the frequency domain (0% error using highest correlated Daubechies-4 Wavelet coefficients of the amplitude spectrum vs. 17.2% error using highest correlated Fourier coefficients of time domain data).

6.5 Random data

The feature selection step is not included in cross-validation although being supervised and therefore being potentially prone to overfitting, i.e. possible overfitting in feature selection will not be suppressed by cross-validation of the classifier. To estimate the amount of overfitting the set of algorithms explained above is applied to randomized data. This dataset was created by randomly shuffling

cycles and measurement points of the TCO gas sensor measurement and assigning each cycle randomly to one of three classes. The cross-validation errors of all combinations of the four feature extraction and two feature selection algorithms described above are distributed around the mean value of 66% with a standard deviation of 1.5%. The expected value for randomly guessing class membership of three classes based on random data is 66.7%; thus, the amount of overfitting in feature selection is small but might increase for larger datasets. Nevertheless feature selection is not cross-validated to save computation time and to make results comparable to results reported in previous publications. Cross-validation of feature selection is a work in progress.

7 Conclusion

As demonstrated the suggested set of methods can be applied to a wide range of different applications. In all four use cases considered in this paper at least one of the suggested methods was able to capture the essential information contained in the cyclical data leading to excellent and often even perfect classification results despite the rather simple classification algorithm used. This again emphasizes the importance of meaningful data representation. Like every other method, this set of methods does not guarantee finding a good data representation but offers a high probability to work well for real world datasets. Nevertheless it is a useful tool for predictive maintenance of complex machines and plants based on sensor networks or development of smart sensor systems compensation disturbing interferences like aging or poisoning of sensor element.

References

1. N. Helwig, E. Pignanelli, and A. Schütze, "Condition Monitoring of a Complex Hydraulic System using Multivariate Statistics," in *Proc. I2MTC-2015*, 2015, pp. 2–7.
2. N. Helwig, E. Pignanelli, and A. Schütze, "Detecting and Compensating Sensor Faults in a Hydraulic Condition Monitoring System," in *Proceedings SENSOR*, 2015, pp. 641–646.
3. K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When Is 'Nearest Neighbor' Meaningful?," in *7th International Conference on Database Theory – ICDT'99*, 1999, pp. 217–235.
4. T. Schneider, "Methoden der automatisierten Merkmalextraktion und -selektion von Sensorsignalen," Master-thesis, Lab for Measurement Technology, Saarland University, 2015.
5. R. T. Olszewski, "Generalized feature extraction for structural pattern recognition in time-series data," PhD-thesis, Carnegie Mellon University, 2001.
6. S. Wold, K. H. Esbensen, and P. Geladi, "Principal component analysis," *Chemom. Intell. Lab. Syst.*, vol. 2.1, pp. 37–52, 1987.
7. Mathworks, "MATLAB Documentation: barttest." [Online]. Available: http://de.mathworks.com/help/stats/barttest.html. [Accessed: 30-Dec-2015].
8. F. Morchen, "Time series feature extraction for data mining using DWT and DFT," *Tech. Rep. No. 33, Dep. Mathmatics Comput. Sci. Philipps-University Marburg, 2003*, pp. 1–31.
9. M. E. Houle, H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Can Shared-Neighbor Distances Defeat the Curse of Dimensionality??," in *21st International Conference on Scientific and Statistical Database Management (SSDBM)*, 2010, pp. 482–500.
10. J. Weston, F. Pérez-Cruz, O. Bousquet, O. Chapelle, A. Elisseeff, and B. Schölkopf, "Feature selection and transduction for prediction of molecular bioactivity for drug design," *Bioinformatics*, vol. 19, no. 6, pp. 764–771, 2003.
11. A. Rakotomamonjy, "Variable Selection Using SVM-based Criteria," *J. Mach. Learn. Res.*, vol. 3, pp. 1357–1370, 2003.
12. S. Abe, *Support Vector Machines for Pattern Classification*, 2nd Editio. London: Springer, 2010.
13. I. Kononenko and S. J. Hong, "Attribute selection for modelling," *Futur. Gener. Comput. Syst.*, vol. 13, no. 2–3, pp. 181–195, 1997.
14. M. Leidinger, T. Sauerwald, W. Reimringer, G. Ventura, and A. Schütze, "Selective detection of hazardous VOCs for indoor air quality applications using a virtual gas sensor array," *J. Sensors Sens. Syst.*, vol. 3, pp. 253–263, 2014.
15. M. Schüler, T. Schneider, T. Sauerwald, and A. Schütze, "Impedance spectroscopy for detection of HMDSO poisoning in metal oxide gas sensors," *Sensors Actuators B, Unpubl.*
16. N. Helwig, S. Klein, and A. Schütze, "Identification and quantification of hydraulic system faults based on multivariate statistics using spectral vibration features," *Procedia Eng.*, vol. 120, pp. 1225–1228, 2015.

Bionotes

Tizian Schneider
Saarland University, Dept. of Mechatronics Engineering, Lab for Measurement Technology, P.O. Box 15 11 50, 66041 Saarbruecken, Germany; and ZeMA – Center for Mechatronics and Automation Technology gGmbH, Eschberger Weg 46, Buisness Park, Building 9, 66121 Saarbrücken, Germany
t.schneider@zema.de

Tizian Schneider studied Microtechnologies and Nanostructures at Saarland University and received his Master of Science degree in January 2016. Since that time he has been working at Centre for Mechatronics and Automation Technology (ZeMA), division 'Sensors and Actuators', in the field of data-based condition monitoring for industrial applications such as fluid power and electromechanical drive systems.

Nikolai Helwig
ZeMA – Center for Mechatronics and Automation Technology gGmbH, Eschberger Weg 46, Buisness Park, Building 9, 66121 Saarbrücken, Germany

Nikolai Helwig studied Mechatronics and received his diploma degree in 2013. Since that time he has been working at Centre for Mechatronics and Automation Technology (ZeMA), division 'Sensors and Actuators', in the field of data-based condition monitoring for industrial applications such as fluid power and electromechanical drive systems.

Andreas Schütze
Lab for Measurement Technology, Saarland University, 66123 Saarbrücken, Germany, and Center for Mechatronics and Automation Technology (ZeMA), 66121 Saarbrücken, Germany

Andreas Schütze received his diploma in physics from RWTH Aachen in 1990 and his doctorate in Applied Physics from Justus-Liebig-Universität in Gießen in 1994 with a thesis on microsensors and sensor systems for the detection of reducing and oxidizing gases. From 1994 until 1998 he worked for VDI/VDE-IT, Teltow, Germany, mainly in the fields of microsystems technology. From 1998 until 2000 he was professor for Sensors and Microsystem Technology at the University of Applied Sciences in Krefeld, Germany. Since April 2000 he is professor for Measurement Technology in the Department of Mechatronics at Saarland University, Saarbrücken, Germany and head of the Laboratory for Measurement Technology (LMT). His research interests include microsensors and microsystems, especially intelligent gas sensor systems for security applications.

5.3 Paper 2: Industrial Condition Monitoring with Smart Sensors Using Automated Feature Extraction and Selection

The following paper extends the previously introduced automated machine learning approach. It showcases its ability to quantify faults, trace features to individual sensors, and detect faults that are not well detectable by relying on engineering domain knowledge.

In comparison to the previous article, the automated machine learning toolbox is improved in the following three ways:

- Feature extraction by ALA and PCA are fitted with a preceding resampling step that provides the necessary scalability in the time series length, which the algorithms themselves cannot offer due to their complexity $O(N^2)$.
- Multivariate RELIEFF replaced the previously univariate RELIEFF to account for feature interactions.
- Statistical moments extracted from equally spaced signal segments were added as feature extractors to allow information extraction from the data's statistical distribution. This addition highlights the automated machine-learning toolbox's modular character, allowing to adapt it as needed.

First, quantifying different machine and sensor faults showcases the toolbox. This demonstration included multiple different faults. Also, the discussed plausibilisation technique based on correct interpolation between training groups is shown in the same use case.

A defective spindle of an electromagnetic cylinder is used as a second example to showcase defect detection, defect localization, and speed detection. This example also shows the toolbox's ability to identify fault symptoms that are counterintuitive to domain knowledge.

Lastly, a lifetime experiment of an electromagnetic cylinder comprised of ~11TB of data is used to show the toolbox in a remaining useful lifetime scenario, its big data capabilities, and its ability to trace features to their individual sensors and frequency ranges.

Industrial Condition Monitoring with Smart Sensors Using Automated Feature Extraction and Selection

Tizian Schneider[1], Nikolai Helwig[1], and Andreas Schütze[2]

[1]*Centre for Mechatronics and Automation Technology (ZeMA gGmbH), Saarbruecken, Germany*

[2]*Saarland University, Lab for Measurement Technology, Saarbrücken, Germany*

Measurement Science and Technology (2018), 29 (9), 094002

The original paper can be found online at https://doi.org/10.1088/1361-6501/aad1d4.

© Used with permission of IOP Publishing Ltd, from Industrial Condition Monitoring with Smart Sensors Using Automated Feature Extraction and Selection, Schneider, Tizian; Helwig, Nikolai; Schütze, Andreas, 29, 9, 2018; Permission was conveyed through Copyright Clearance Center, Inc.

IOP Publishing Measurement Science and Technology
Meas. Sci. Technol. **29** (2018) 094002 (15pp) https://doi.org/10.1088/1361-6501/aad1d4

Industrial condition monitoring with smart sensors using automated feature extraction and selection

Tizian Schneider[1], **Nikolai Helwig**[1] **and Andreas Schütze**[2]

[1] ZeMA, Center for Mechatronics and Automation Technology gGmbH, Saarbruecken, Germany
[2] Department of Mechatronics, Lab for Measurement Technology, Saarland University, Saarbruecken, Germany

E-mail: t.schneider@zema.de

Received 15 February 2018, revised 28 May 2018
Accepted for publication 6 July 2018
Published 1 August 2018

Abstract

Smart sensors with internal signal processing and machine learning capabilities are a current trend in sensor development. This paper suggests a set of complementary and automated algorithms for feature extraction and selection to be used with smart sensors. The suggested methods for feature extraction can be applied on smart sensors and are capable of extracting signal characteristics from signal shape, time domain, time-frequency domain, frequency domain and signal distribution. Feature selection subsequently is capable of selecting the most important features for linear and nonlinear fault classification. The paper also highlights the potential of smart sensors in combination with the suggested algorithms that provide both data and further functionality from self-monitoring to condition monitoring in industrial applications. The first example applications are condition monitoring of a complex hydraulic machine where smart signal processing allows classification and quantification of four different fault scenarios. Additionally redundancies in the systems were used for self-monitoring and allowed to detect simulated sensor faults before they become critical for fault classification. The second example application is remaining lifetime prediction of electromechanical cylinders that shows applicability to big data and transparency of the solution by providing detailed information about sensor significance.

Keywords: feature extraction, feature selection, condition monitoring, smart sensors

(Some figures may appear in colour only in the online journal)

1. Introduction: state of the art and current trends

A current trend in sensor technologies is the integration of additional functionality by use of active measurement principles as seen in various sensor elements and systems. Examples include

- Magnetic sensors like Hall sensors that use spinning-current, internal calibration and even correction of offset temperature coefficients though integration of internal chip heaters [1] or magnetoresistive (MR) sensors using the compensation principle to suppress temperature cross sensitivity [2];
- Micromechanical accelerometers (also using compensation or internal calibration methods) [3] and gyroscopes (using the Coriolis effect with active vibration excitation) [4];
- Coriolis-based flow sensors also using active excitation and determination of the resonance frequency to measure the density of gases or fluids [5];
- Chemical sensors using temperature or gate bias modulation for improved selectivity, sensitivity and stability [6].

Active modes of operation also offer additional potential for self-diagnosis, which is already extensively being used in automotive applications [7]. This does not only apply to inertial sensors, where the correct function is checked with

 © 2018 IOP Publishing Ltd Printed in the UK

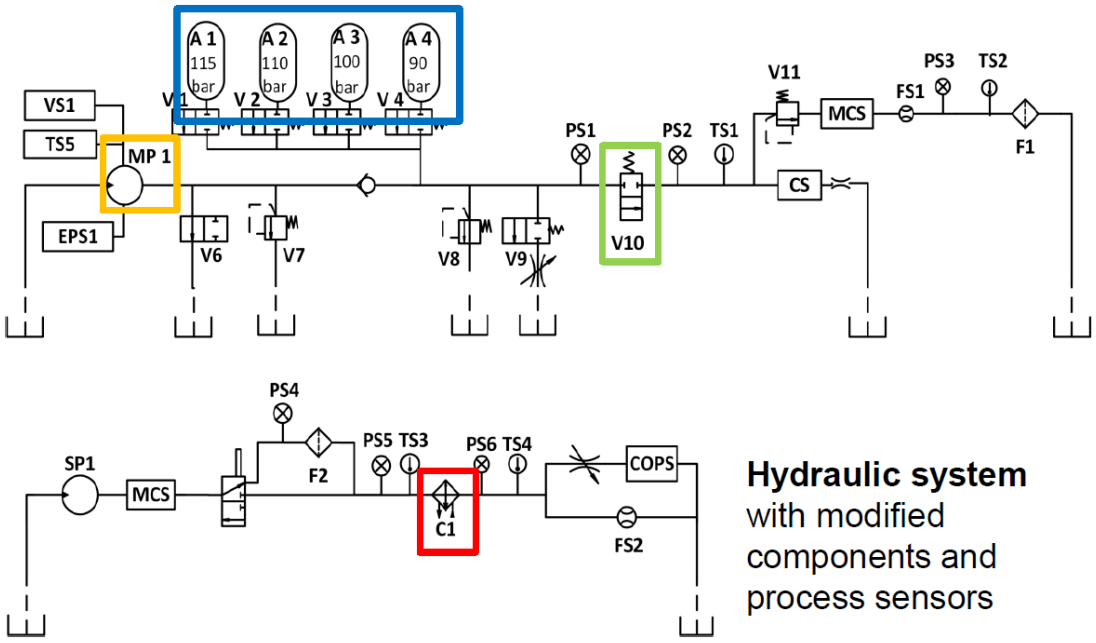


Figure 1. Hydraulic system for collection of training data: (top) working circuit with main pump MP1 (orange) with switchable orifice V9, switchable accumulators A1–A4 with different precharge pressures (blue) and variable load V10 (green); (bottom) cooling and filtration circuit with cooler C1 (red) [12]. All marked components can simulate faults with different steps of severity: the pump, for example, can simulate internal leakage by a switchable bypass.

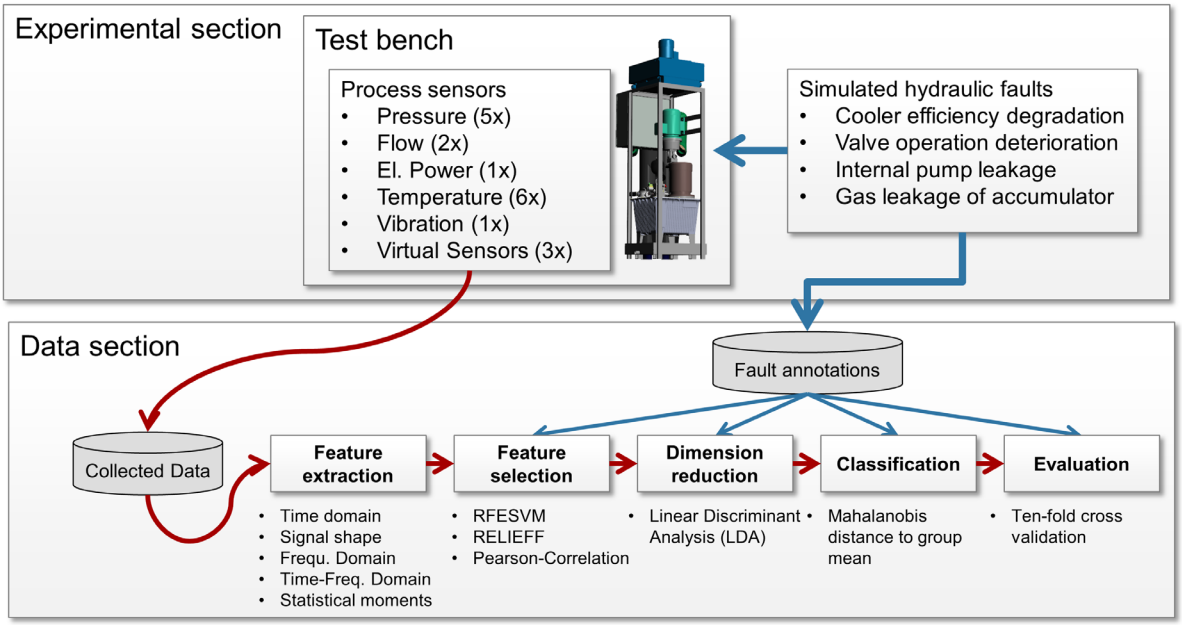


Figure 2. iCM-hydraulic experimental and MoSeS-Pro data analysis approach. (top) Experimental setup with multiple simulated faults and process sensors to read sensor responses under different fault conditions. (bottom) Data analysis with gradual dimensionality reduction (feature extraction, selection and LDA), machine learning (classification) and cross-validation. Reproduced with permission from [16].

internal excitation, but also for e.g. the lambda probe: here the time constant for heating to the desired operating temperature is used to detect faults, such as cracks of the ceramic. Self-diagnosis is especially important for applications in safety and security. Fire detection and explosion protection could not be addressed with low-cost gas sensors, which are prone to poisoning. Here, dynamic operation also allows detection of sensor faults, e.g. poisoning of the sensor material [6].

Thus, smart sensors with additional functionality provide a significant added value for higher-level functions, e.g. in production systems. The correct sensor function is also required for condition monitoring of complex systems (see section 2 below). In this case, the correlation of sensor data within the system can also be used to verify the correct sensor function; however, in this case, the sensor fault diagnosis has to be performed at a higher level within the system.

Additional trends that will be initiated or at least pushed further by the Industry 4.0 paradigm are

- Measurement as a service: this could be a trend similar to the service provided by Uber in public transport, i.e. measurement values are sold instead of instruments [8]. Note that the measurement uncertainty—determined online by self-calibration—will then influence the price.
- Traceability of individual components down to screws, individual gears and even gaskets: this additional knowledge will allow tolerance measurement in the assembly of (sub-)systems and is also required for a comprehensive condition monitoring to assess the influence of individual processing steps and machines on the final result.
- Self-learning systems: the correlation between sensor data as well as other process and ambient parameters can be evaluated to ensure the correct function of the system in the sense of a system self-diagnosis by making use of machine learning [9]. So far it is unclear if unsupervised methods are sufficient or if supervised learning is required, i.e. knowledge of the current system status for training the evaluation.
- Semantic technologies for analysis of complex systems: interpretation of measurement values beyond the purely data-based approaches could offer further opportunities, e.g. for plausibility checks of sensor data and for providing confidence values for (fault) causes. Note that the World Wide Web consortium (W3C) has started working on a semantic sensor network ontology as early as 2005 which allows representation of measurement values and their significance [10].

The last example shows that the importance of sensors and measurement technology was recognized also by other parties, which leads to some parallel and independent developments. Interestingly, however, aspects like measurement uncertainty and sensor self-monitoring are not addressed in the context of semantic technologies even though semantic representation would be highly valuable especially for these aspects [11].

2. Condition monitoring using data-based modeling

The potential of data-based sensor signal evaluation is demonstrated by the projects intelligent condition monitoring (iCM) Hydraulics and MoSeS-Pro (modular sensor systems for real time process control and smart condition monitoring). In iCM Hydraulics a hydraulic model system combining a primary circuit with variable load and a secondary circuit for cooling and filtration was used to study the identification of typical system faults (internal pump leakage, delayed valve switching, pressure leakage in the accumulator, reduced cooling efficiency) only based on an evaluation of the typical process sensors (pressure flow rate, temperature, electrical power). The schematic of the test system is shown in figure 1.

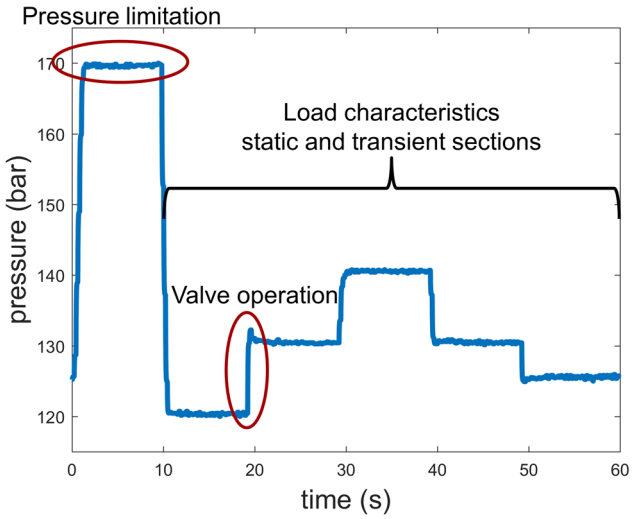


Figure 3. Fixed working cycle (measured by PS1) with pre-defined load steps with static and transient sections. During the first 10 s V10 (see figure 2) is switched off and the system runs into the maximum pressure limitation. During the following 50 s the system simulates a hydraulic press application by setting different pressure levels using valve V10.

Figure 2 provides an overview of the approach: the hydraulic system is equipped with a total of 18 physical and virtual (e.g. efficiency calculated from electrical power input and hydraulic power output) sensors, which are read-out with up to 100 Hz.

The system was used to simulate a periodic industrial process with a working cycle of 1 min duration that is shown in figure 3. In each cycle a total of approx. 50.000 raw sensor values is recorded, which are interpreted as a high-dimensional measurement vector. A multi-step dimensionality reduction covering feature extraction and selection yields a projection obtained by linear discriminant analysis (LDA) [13], which allows classification of the system status, i.e. identification and quantification of the fault. Classification is performed based on the Mahalanobis distance of measured vectors to training group centers. Note that feature extraction is realized with unsupervised methods, i.e. without making use of the system status, while feature selection—here based primarily on support vector machines—and LDA projection are supervised methods, i.e. require the knowledge of the system status [14]. The evaluation is based on a comprehensive training phase in which all combinations of all fault states are tested. The complete training is based on several 1000 working cycles and requires approx. 3 d, primarily due to the relatively slow equilibration of the temperature after changing the cooling efficiency. The complete training data set contains almost 120 Mio raw data points. A systematic validation, e.g. based on k-fold cross-validation and projected faults, completes the development of the statistical model and ensures that no overfitting occurs in spite of the high-dimensional input data set and the supervised training methods [14].

In this example features were extracted from time domain using adaptive linear approximation (for accumulator pressure loss, internal pump leakage and valve faults) and from time-frequency domain using Best Daubechies Wavelets (for cooler

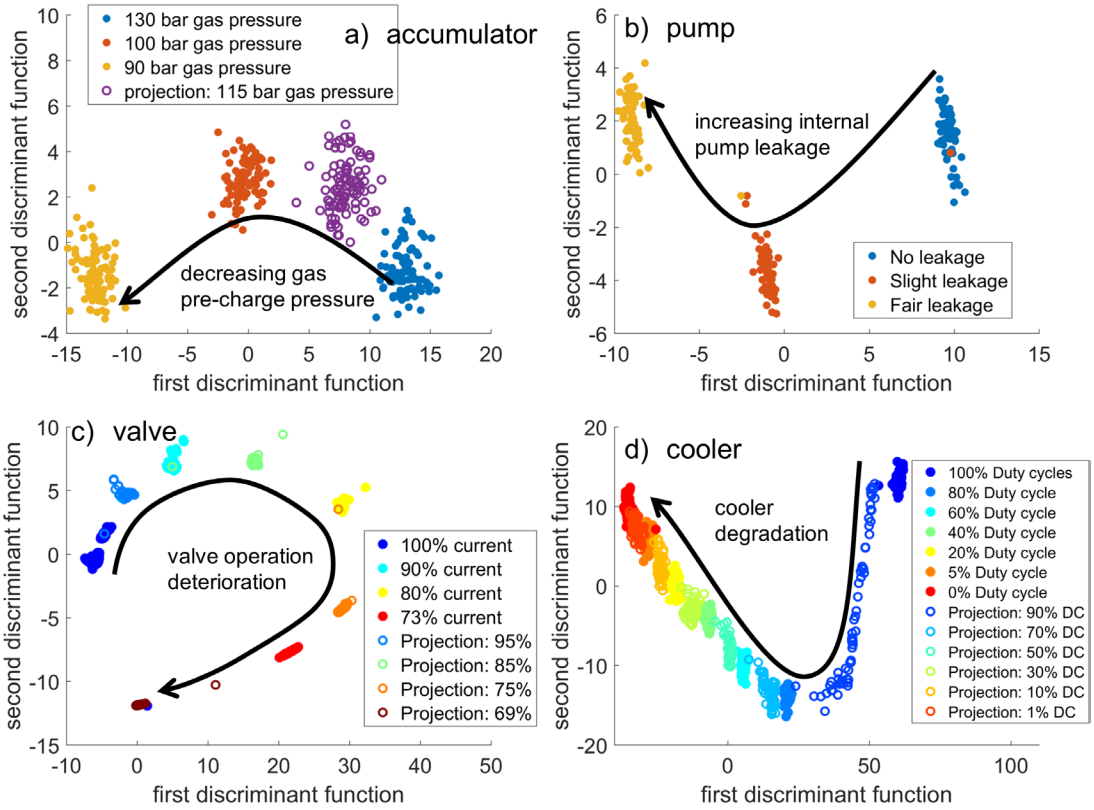


Figure 4. Results for determination of the four system faults studied: accumulator pressure (a), internal pump leakage (b), valve operation (c) and cooler degradation (d). Full symbols show data used for determining the statistical model, open symbols show additional test data not used in the training which prove that unknown data are interpreted correctly [14], as unknown data are always projected in between the correct adjacent training data of the corresponding fault, i.e. they prove the correct interpolation achieved with the data-processing.

faults). The methods are explained in detail in section 3. Both methods can be implemented very efficiently on low cost hardware. They require less than 1 min on a standard PC for the complete data set with several 1000 cycles. The resulting maximum of 500 features result in a feature space that still has too many dimensions for efficient classification. Therefore, feature selection based on recursive feature elimination support vector machines (for details see section 3.2) for each target, i.e. fault type, is used which is the computationally most expensive step and takes several minutes on a standard PC. Note that these algorithms are chosen automatically as described in section 3.3. The pressure loss of the accumulator was detected using 21 features, 30 for cooler degradation, 10 for internal pump leakage and two for valve operation deterioration. The subsequent calculation of the LDA projection to obtain the 2D plots, see figure 4, or ideally only one discriminant function per system fault only takes a fraction of a second on the same hardware. Even faster is the classification of a new working cycle, i.e. extraction of the selected features, projection in the LDA space for each system fault and classification based on a Mahalanobis classifier, which can thus be performed in real-time even on a low-cost microcontroller-based system.

The performance of the approach is demonstrated in figure 4 for the four studied system faults: each fault state can be identified independently and its severity or level can be estimated with surprisingly high accuracy. The cooler efficiency, for example, can be estimated with an accuracy of better than 10% (the reduced cooler efficiency was simulated with pulse

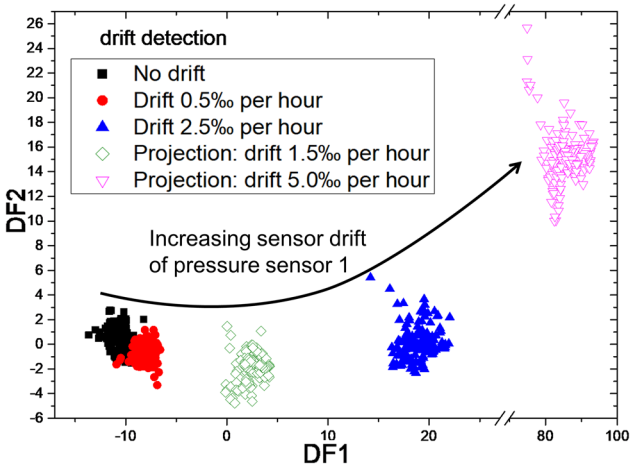


Figure 5. Results for identification of sensor drift. Full symbols show data used in determining the model for the sensor fault diagnosis, open symbols show additional data not used in the training to prove that unknown data are interpolated and, in this case, even extrapolated correctly [17]. © 2015 IEEE. Reprinted, with permission, from [12].

width modulation of the power supply, the percentage gives the duty cycle used); the accumulator pressure can be determined with an uncertainty of approx. 10 bar. Projected test data which were not used to build the model (open symbols) show that the model allows correct classification of unknown states and even an extrapolation of data outside the training range is possible within limits.

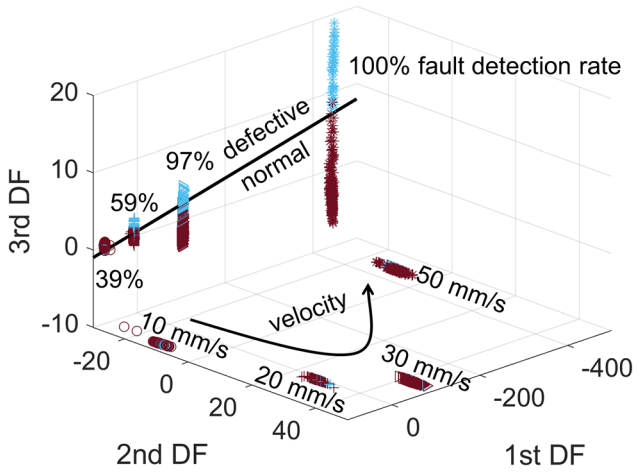


Figure 6. LDA projection of 30 automatically selected vibration features, with training based on velocity classes 10, 20 and 50 mm s^{-1}; classification rate determined for Mahalanobis distance classifier with 10-fold cross-validation. The velocity class 30 mm s^{-1} projected with the model again proves the shows correct interpolation achieved with the model.

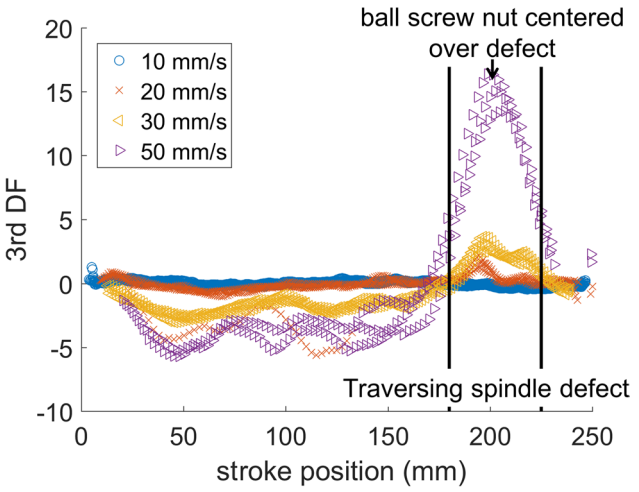


Figure 7. Deliberate abrasion as local defect on the spindle and corresponding signal of DF3 versus stroke position (moving average over 10 data points) [16]. A simple threshold for the value of DF3 is sufficient to locate the defect position during the stroke. The identification becomes more obvious as the velocity is increased.

In further experiments we could show that the training can be transferred from one system to a second, identical system after some calibration, i.e. shift of the LDA projections for the correct system state [15]. Because of the high performance, which was not expected when designing the experiments, we also studied how sensor faults would influence the classification results. Typical sensor faults, namely offset, drift, noise and signal drop-outs were simulated in the recorded data for all sensor channels and the resulting data were used to classify the system state. Not surprisingly, the classification rate is drastically reduced, especially for monitoring of pump leakage and hydraulic accumulator. The sensor faults were defined as new targets for the classification algorithm to allow automatic recognition and train using the same completely automated approach. Again, the simulated sensor faults could be recognized with high reliability independent of the system state as shown in figure 5 for the simulated drift of a pressure sensor. In fact, sensor faults can be diagnosed before they lead to false classification of the system state [16]. Correct classification of the overall system state is still possible by excluding the defective sensor(s) from the evaluation and evaluating the remaining sensors. In fact, up to five of the most important sensors can be excluded from the evaluation and still a correct classification rate of more than 80% is achieved for the various system faults [16].

Another example for data-driven modeling is shown in figures 6 and 7. A miniaturized sensor system prototype was designed for integration in an electromechanical cylinder (EMCs). These are increasingly applied as feed drives in machine tools, due to their unique combination of high loads, precision, and flexibility. The sensor system combines various (partially redundant) sensors: linear and rotary encoders, 3D accelerometers, microphone, temperature and IR radiation sensors. Currently, the sensor prototype consists of two separate subsystems: First, two stacked sensor PCBs mounted on the front surface of the ball screw inside the EMC housing (Festo ESBF-BS-63-400-5P, ⌀ 63 mm, 400 mm stroke, 5 mm spindle pitch, axial load max. 7 kN) carrying a total of nine MEMS sensors. Furthermore, the rotary position of the spindle shaft is measured by an AMR Wheatstone bridge sensor [17] with an external bias magnet generating the support field which interacts with the ferromagnetic teeth of the spindle shaft. This sensor is positioned at a fixed position in the cylinder housing close to the ball bearing pointing to the thread with a working distance of 1 mm. During rotation, the relative position of sensor and teeth changes periodically resulting in sine and cosine sensor signals. Note that despite the fact that condition monitoring of EMCs and hydraulic machines are very different the data structure of time series that can be treated as working cycles (actual working cycle for the hydraulic machine; sliding windows in case of EMCs) are identical and therefore can be treated using the same data processing approach. The diversity of the datasets thereby shows the versatility of the approach.

To evaluate the sensor system in a condition-monitoring scenario, we induced a local abrasion of the spindle at stroke position 185 mm and recorded several stroke movements with varying velocity and three repetitions. Short-time Fourier transform (STFT) was applied (length 10 000/overlap 2000 samples) for signal processing with subsequent feature extraction and selection as previously demonstrated [18]. Feature extraction captures a total of 210 statistical parameters such as median, variance, skewness, and kurtosis in different intervals of the amplitude spectra of three acceleration axes (FXLN sensor). The features are selected by *F*-value ranking of univariate ANOVA and dimensionally reduced to three discriminant functions (DFs) using linear discriminant analysis (LDA) to obtain the maximum class separation. The latter algorithms are supervised learning methods, i.e. require class-annotated data which were given as velocity information and local spindle condition traversed by the spindle nut. Figure 6 shows the resulting 3D-projection of sensor data with the planes DF1–DF2 and DF1–DF3 separating the different velocity levels and spindle conditions, respectively. Here, the velocity classes with 10,

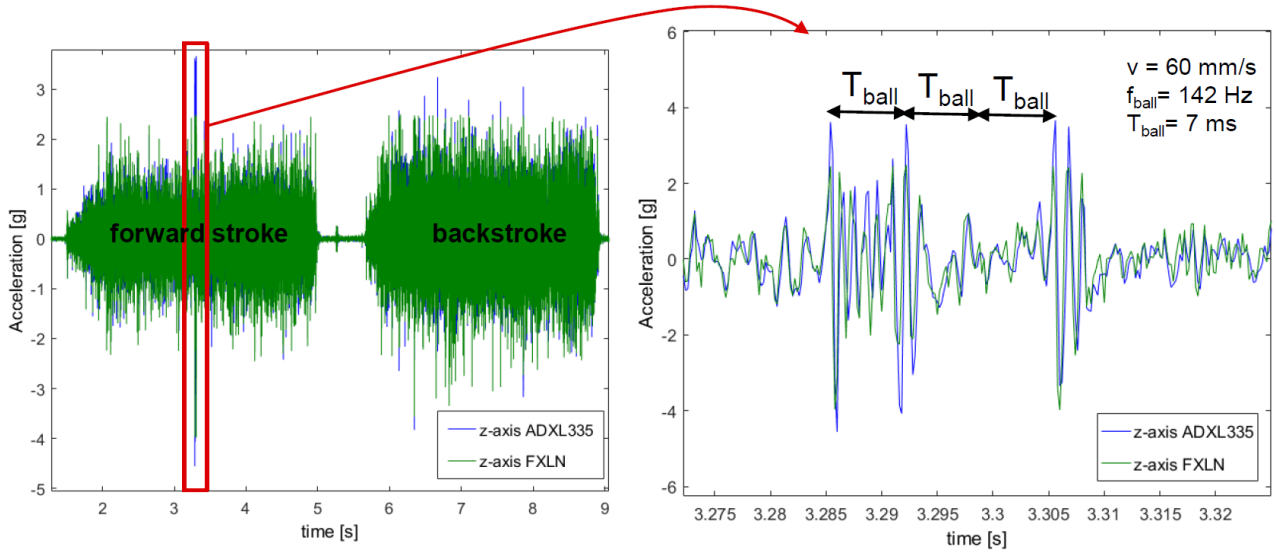


Figure 8. (left) Acceleration signal of forward stroke and backstroke in time domain traversing the prepared thread defect at 200 mm (total stroke is 400 mm). (right) Zoom in on defect. T_{ball} is the inverse theoretical ball pass frequency. Only few balls seem to interact with the thread defect [16]. Furthermore the fault, i.e. the interaction of balls and thread defect, is only observed during the forward stroke due to different contact angles.

20, and 50 mm s^{-1}, respectively, were used for training and the class with 30 mm s^{-1} velocity was used for evaluation. The intermediate velocity class fits well into the data-based model and the fault identification rate improves with increasing velocity. Figure 7 shows the plot of DF3 over the stroke position clearly indicating the defect. The maximum is blurred, first, due to the interaction of balls and spindle defect over a distance of 30 mm and, second, also results from the STFT temporal blur. Furthermore, especially at low speeds with accordingly higher local resolution, two local maxima can be seen indicating the entry and exit points of the spindle nut passing over the defect. This example shows that the stroke position dependent analysis of signals can be used for fault diagnosis differentiating between local anomalies such as defects of the spindle and global disturbances, e.g. of the bearings in the ball screw or of the ambient. Further details can be found in [18].

As shown in figure 8 the vibration sensors ADXL335 and FXLN measure sharp axial acceleration peaks when balls traverse the local flank defect that can be seen in time domain signal. The peaks are only visible in the signal recorded during the forward stroke, which can easily be explained by different contact angles on forward and back stroke. On the other hand, the observed effect, that only a few balls in the ball screw nut interact noticeably with the thread defect, makes prediction based on a physical model complicated. The pattern observed in figure 8, right, fits the expected ball pass frequency of 142 Hz but is not deterministic, i.e. not every ball passing over the defect leads to an acceleration peak. Nevertheless, the automated pattern recognition approach is able to identify this pattern and therefore can discriminate between good and defective axes (see figure 6).

3. Fully automated modular algorithm toolbox

The successful preliminary work in iCM Hydraulics resulted in the establishment of the successor project MoSeS-Pro [19],

Table 1. Advantages and disadvantages of ALA.

Advantages	Disadvantages/assumptions
•Extracts information from local details in time domain	•Assumes multiple linear signal segments within the cycle
•Linear function provides first order approximations	•Assumes the same splits can be performed on all cycles
•Noise suppression	•Signal length has to be limited due to computational complexity
•Does not create new clusters within the data	

in which the developed methods are transferred to an open sensor system toolbox. In this project magnetoresistive sensors (anisotropic magnetoresistive (AMR), giant magnetoresistive (GMR) and especially tunneling magnetoresistive (TMR)) are primarily used as they are versatile tools for measuring current, position and angle yielding periodic sine-cosine-signals. In addition, other MEMS (micro-electro mechanical system) sensors, e.g. for noise, vibration, pressure or thermal radiation, are used to extend the measurement spectrum. All sensor principles are also integrated into components and subsystems [20] to allow improved performance and condition monitoring of mechatronic components, both as end-of-line test in their production and during their operation in manufacturing systems. In MoSeS-Pro, modular electronics and software algorithms are developed allowing the required signal pre-processing and feature extraction directly in the smart sensor system. Otherwise, signals recorded at high frequencies above several 100 MHz would result in data rates which would overload the higher-level systems in a typical production environment. In addition, novel self-*X* methods, wireless sensor interfaces and energy harvesting are developed for easy integration and initialization of system operation.

To make statistical data analysis a powerful tool for condition and process assessment without firm and detailed expert

Table 2. Advantages and disadvantages of PCA.

Advantages	Disadvantages/assumptions
•Best linear transformation in terms of low approximation error	•Optimal performance only, if signal can be decomposed into multiple linear driving forces
•Good representation of general signal shape	•Local details are neglected
•Noise suppression	•Signal length has to be limited due to computational complexity
•Does not create new clusters within the data	

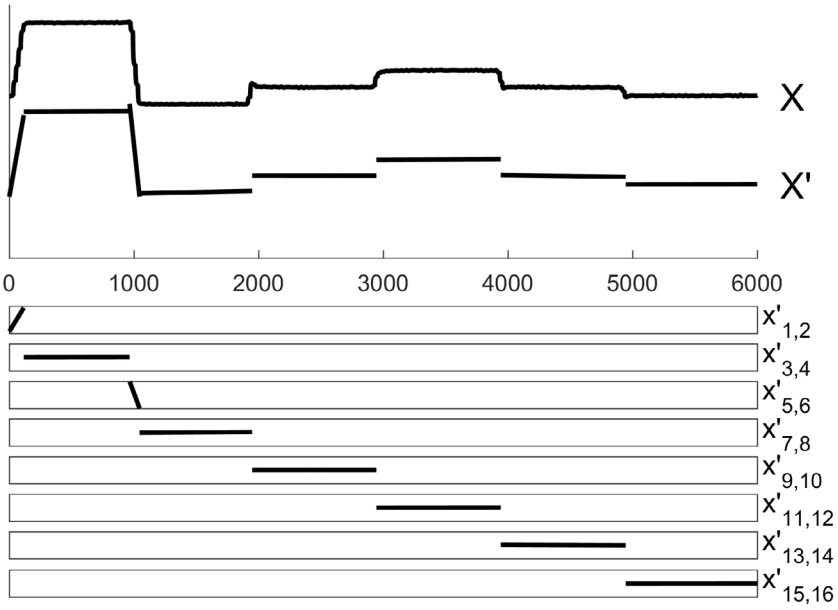


Figure 9. Approximation X' using ALA with features $x'1$–$x'16$ compared to the original sensor signal X (here a pressure sensor over a complete working cycle, shifted for better clarity). The eight segments shown below are represented by mean values (uneven indices) and slopes (even indices).

knowledge, the underlying algorithms need to be self-optimizing and combined in automated signal processing chains. Since no machine learning algorithm can guarantee optimal results on all datasets a toolbox of automated algorithms is used that was designed to solve typical problems faced when applying machine learning for condition monitoring. However, this approach, especially in the case of supervised learning, requires a sufficient quality of training data, i.e. typically process-synchronized time series sensor data which are annotated with corresponding classes, i.e. the target vector for which the statistical model is to be trained. The typical steps for offline analysis are signal pre-processing, feature extraction (FE) and feature selection (FS) as well as classification with subsequent evaluation and can be interpreted as a gradual dimensionality reduction. Feature extraction and selection can be fully automated using the developed modular approach based on complementary algorithms to extract information that is usually used for remaining useful lifetime estimation and fault classification. Such information is typically extracted using PCA [21, 22], Fourier transformation [23] or wavelet transformation [24]. Similarly, complementary techniques are also used to select suitable features and feature combinations, i.e. by a simple analysis of the signal correlation with the target condition or by recursive feature elimination support vector machines (RFESVM) for linear or RELIEFF for nonlinear class separability [25]. In this way, the signal processing software as part of the sensor kit is realized in a highly modular design since heterogeneous sensors differ significantly regarding signal shape, time and spatial resolution, as well as target information to be extracted. Most important, however, all of these algorithms are simple enough to be integrated directly in the smart sensor using either a microprocessor or, for signals with higher data rates, an FPGA board. The individual algorithms are introduced in the following section.

3.1. Automated feature extraction

All FE methods introduced here are unsupervised and aim to reconstruct the original raw data with as few features as possible. Simultaneously each method has to preserve similarities within classes, i.e. machine conditions, and differences between classes to find good representations for machine learning. Of course, there is a trade-off between low approximation error and low number of features. Furthermore, the method that extracts the features that provide highest contrast between different classes cannot be determined beforehand. Therefore multiple simple but complementary methods are tested and evaluated to identify the best one. The following methods are used for FE:

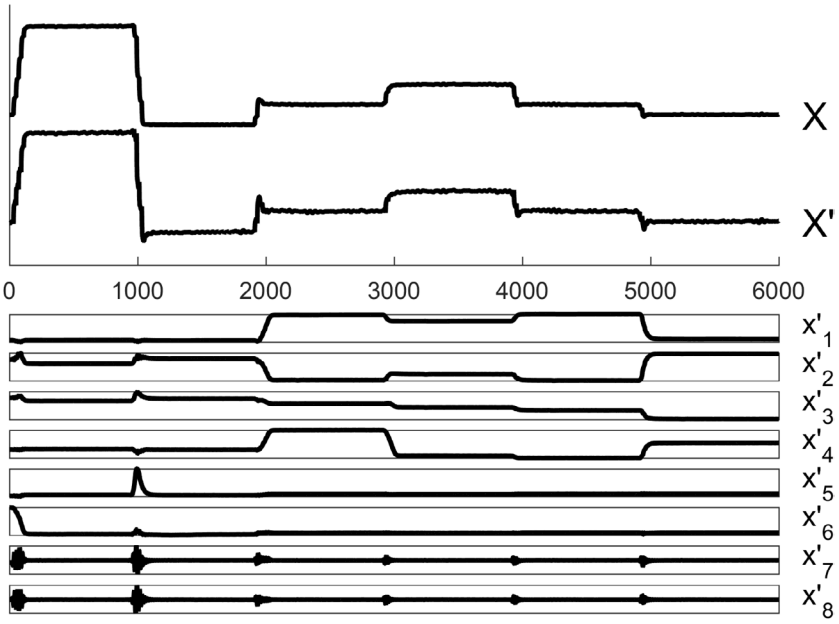


Figure 10. Approximation X' of X using PCA (shifted for better clarity). The extracted features $x'1$–$x'8$ are projections onto the first eight principal components that are shown below. The reconstruction is achieved by the sum of the PCs weighted with the corresponding coefficient (feature).

Table 3. Advantages and disadvantages of BFC.

Advantages	Disadvantages/assumptions
•Extracts information from local details in the frequency domain	•Assumes information to be located in frequency domain
•Takes into account phase shifts	•Fixed data reduction factor
•Low computational complexity ($n\log n$)	•Prone to frequency shifts
	•FFT performed on complete signal length blurs frequencies from individual segments

Adaptive linear approximation (ALA): ALA splits the measurement time segment or working cycle into variable length, approximately linear segments and extracts slope and mean value of each segment, as shown in figure 9. Start and end of each segment are chosen to minimize the overall reconstruction error over all training cycles and all cycle segments for a given number of splits performed on the data. Thereby the number of splits that controls the trade-off between low approximation error and low number of features is chosen automatically by monitoring the decrease of approximation error when performing one additional split. If the error does not decrease significantly with additional splits the major characteristics of the cycle shape are represented by the features and the algorithm stops. A detailed description of the algorithm can be found in [26]. This algorithm has been chosen over multiple other variants [27–29], because it guarantees to perform splits with lowest approximation error. The disadvantage of this method is the high computational complexity of $O(n^2)$ with the number of data points n per time segment. In practice, this requires the maximum signal length to be limited to 500 values, e.g. by resampling, in order to achieve reasonable computing times. The maximum cycle length has been determined empirically to match the requirements for low computational cost in MoSeS-Pro and the Big-Data application shown in section 4 and at same time not to be restrictive in one of the applications mentioned in this paper. Nevertheless the advantages of a low number of extracted features and good representation of local information make ALA an excellent algorithm for Feature Extraction in time domain. See table 1 for a comparison of advantages and disadvantages of ALA.

The n first principal components (PCs) found by principal component analysis (PCA) are the optimum linear transformation of the signal in terms of minimal approximation error for a given number of features [30]. This is equivalent to PCA disassembling the signal into multiple linear driving forces ordered by descending significance [30]. As well as for ALA the maximum signal length is limited to 500 due to computational complexity $O(n^2)$. Again this limitation is an empirically found trade-off between necessary accuracy and computational cost following the same ideas discussed for ALA [31]. Overall, PCA is optimal for representing the overall signal shape. Nevertheless, local details like sudden changes and edges (see figure 10) might be neglected in the first principal components in case they are not traceable to linear driving forces or do not significantly affect the global approximation error. For a list of advantages and disavantages see table 2. The number of principal components to extract is determined by dividing the maximum number of features FS can efficiently deal with (in our case approx. 500, for explanation see section 3.2) by the number of sensors in the dataset to capture as much information of each sensor as possible.

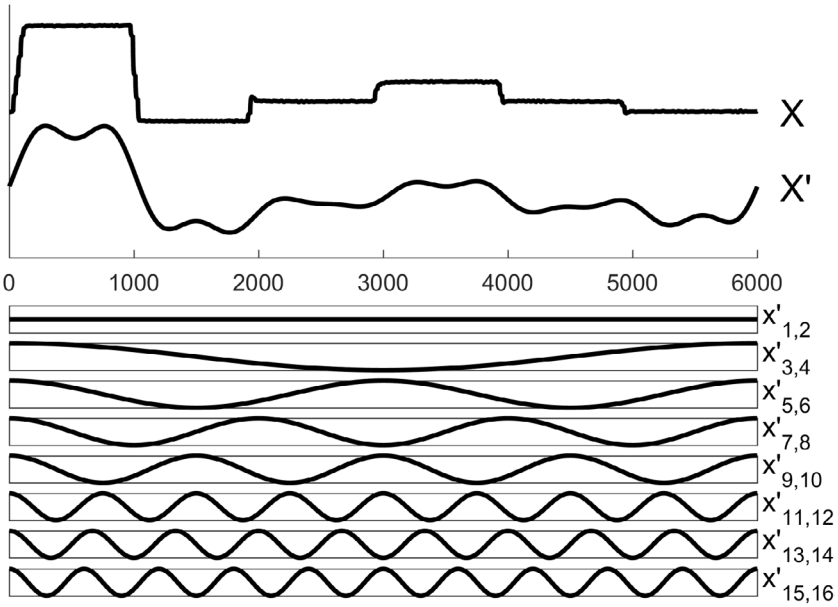


Figure 11. Approximation X' using BFC with features $x'1$–$x'16$ compared to the original signal X (shifted for better clarity). The approximation is the superposition of multiple sine waves represented by amplitude (uneven indices) and phase shifts (even indices). As shown in comparison to figures 9 and 10, in this case approximation in time-domain is more accurate and will be selected automatically (see section 3.3).

Best Fourier coefficients (BFC) extract amplitude and phase of the Fourier coefficients with highest mean absolute value over all cycles (see figure 11). These Fourier coefficients are the ones that contain the most signal energy and therefore contribute most to low approximation error [32]. This algorithm is therefore especially suitable for information that is well localized in the frequency domain, e.g. for vibration signals or motor current analysis. The data reduction factor of this method is fixed to ten, as this was shown to be a reasonable tradeoff between low number of features and low approximation error (see section 3.4). If, after omitting the smaller 90% of the Fourier coefficients, the number of features is still too high for efficient multivariate FS the best 500 features are selected based on the highest absolute Pearson correlation to the target value (see section 3.2). A summarized list of advantages and disadvantages can be found in table 3.

Best Daubechies wavelet coefficients (BDW) extracts the coefficients with highest mean absolute value of a multilevel wavelet transformation into the time-frequency domain using Daubechies-4 wavelets (see figure 12). As for BFC these coefficients contribute most to achieving a low approximation error [32] and the best 10% of the coefficients are extracted. The Daubechies-4 wavelet was chosen because of its widespread use in signal processing and data compression. The fixed data reduction factor is again a trade-off between low number of features and low approximation error that was shown to be reasonable (see section 3.4). Advantages and disadvantages can be found in table 4. If necessary, preselection with Pearson correlation is applied to reduce the number of extracted features to 500 (see section 3.3). BDW captures information in time-frequency domain and provides a multiresolutional view to the data because it is applied multiple times for multi-level wavelet transformation.

For information that is contained in the statistical distribution of the measurement values signals are split into a fixed number of equally sized segments and the first four statistical moments of the data distribution—mean, standard deviation, skewness and kurtosis—are extracted for each segment. For advantages and disadvantages see table 5.

Table 4. Advantages and disadvantages of BDW.

Advantages	Disadvantages/ assumptions
•Extracts information from time-frequency domain •Provides multi-resolutional view for both overall shape and local details •Low computational complexity (n)	•Daubechies-4 wavelet might not fit every data •Fixed data reduction factor

3.2. Automated feature selection

After each of the feature extraction methods is applied to each available (sensor) signal, the features derived with each method are pooled and FS is applied to each of the five resulting feature pools to select a feature subset suitable for the desired classification task. Both this step (including preselection) and the classification step are based on supervised learning and therefore rely on the target value being known for each training signal. The pooling step is also called data fusion since features from multiple sensors are combined. Due to multiple problems that arise for feature selection there is no FS algorithm that will guarantee an optimal feature subset for every classification task other than exhaustive search [33], which is of course computationally prohibitive in most cases. Therefore, the following complementary feature ranking algorithms are used on every feature pool. The optimal size of the feature set to be used is estimated by computing the 10-fold cross-validation classification error (see section 3.3) while

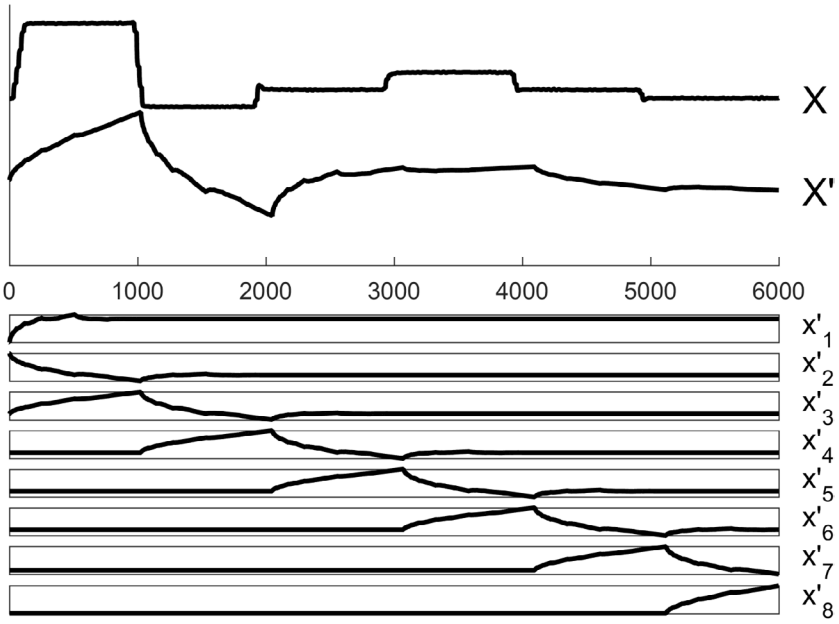


Figure 12. Approximation X' of X using BDW (shifted for better clarity). Extracted features are wavelet coefficients $x'1$–$x'8$. The reconstruction is achieved by the superposition of the wavelets shown below weighted with the corresponding wavelet coefficients (features). As shown in comparison to figures 9 and 10, in this case approximation in time-domain is more accurate and will be selected automatically (see section 3.3).

adding features according to their relevance ranking until 500 features are selected. The set with minimal cross-validation error is chosen for classification. This method guarantees optimal feature subset size and is possible due to the low computational cost of LDA and Mahalanobis distance used for the classification (see section 3.3). The maximum number of 500 features was chosen to be as small as possible to allow fast evaluation of the feature sets and as big as possible to include sufficient information for the classification algorithm. In most applications the training set is too small (usually <1000 cycles) to justify more than 500 features and the minimal classification error is achieved with less than 100 features [34].

Recursive feature elimination support vector machines (REFSVM) is a multivariate technique for FS based on training a linear SVM in each recursion. The normal vector of the found optimal separating hyperplane represents the direction of optimal class separation. The feature that contributes least to this vector, i.e. has lowest absolute value, is eliminated and the algorithm is repeated for the remaining feature set. Details of the algorithm can be found in [35]. Multiclass problems are resolved using one versus one multiclass encoding and computing the mean of all absolute feature weights. The SVM parameter C is set to 1000 and features are standardized to have a mean value of 0 and a standard deviation of 1 before the algorithm is applied. The value of the parameter C defines the penalization of missclassifications and is usually a good tradeoff between generalization and adaptation to the training data. The standardization is necessary to account for different feature scales [36]. RFESVM was shown to be very effective and reliable in a comparison of 66 FS algorithms for gas sensor and condition monitoring data [31]. Nevertheless it relies on the classes being linearly separable. A list of advantages and disadvantages is given in table 6.

Table 5. Advantages and disadvantages of statistical moments.

Advantages	Disadvantages/assumptions
•Extracts information from statistical distribution of data values	•Equally sized segmentation might be meaningless
•Low computational complexity (n)	•Meaning of features is hard to interpret

If the classes are not linearly separable RFESVM is complemented by RELIEFF (fixed name), a multivariate FS algorithm that is based on K-nearest-neighbors and therefore a nonlinear radial classification. RELIEFF finds the k-nearest neighbors for each point of the same group and the k-nearest neighbors of different groups and updates the ranking vector according to the contrast between nearest hits and misses provided by the features. Usually the L1 norm is used as distance metric [37]. In our case k is set to 3 to prevent a highly fractural decision border in case of overlapping groups and low computational cost. A comparison of advantages and disadvantages can be found in table 7.

RFESVM and RELIEFF are extremely powerful methods for feature selection taking feature interaction into account. Nevertheless they internally rely on machine learning and can therefore suffer from overfitting, the ‘curse of dimensionality’ and nonlinear algorithmic complexity if the number of features in the original feature pool is high. When more than 500 features are in the pool, feature interaction is neglected and features are ranked by their individual Pearson correlation to the target value to select the 500 most relevant individual features for RFESVM and RELIEFF. Pearson correlation is thus used both for preselection and for feature selection itself. Preliminary work showed that 500 features are sufficient to solve all feature selection tasks the methods have been applied to and at the

Table 6. Advantages and disadvantages of RFESVM.

Advantages	Disadvantages/assumptions
•Most reliable FS algorithm in a comparison of 66 algorithms [31] •Internally relies on linear SVM and is therefore very robust against overfitting •Takes feature interaction into account •Takes redundancy into account	•Assumes different classes to be linearly separable by a single hyperplane •Needs to be limited to 500 features due to high computational cost •Fixed parameter *C* might be suboptimal for some applications •Ignores nesting effects

Table 7. Advantages and disadvantages of RFLIEFF.

Advantages	Disadvantages/assumptions
•Second-most reliable FS algorithm in a comparison of 66 algorithms [31] •Selects features for radial classification •Broadly used [38, 39] •Takes feature interaction into account	•Needs to be limited to 500 features due to high computational cost •Fixed parameter *k* might be suboptimal for some applications •Ignores nesting effects

Table 8. Advantages and disadvantages of Pearson correlation.

Advantages	Disadvantages/assumptions
•Applicable to huge number of features due to very low computational cost •Results are easy to understand	•Only applicable in case of numerical target •Ignores nesting effects •Ignores feature interaction (assumes some features are relevant by themselves) •Ignores redundancy •Only measures linear correlation

Table 9. Advantages and disadvantages of LDA and Mahalanobis classification.

Advantages	Disadvantages/assumptions
•Allows determination of optimal feature subset size by brute force due to low computational cost. •Offers 2D visualizations •Applicable for regression (with quantification) as long as system response is monotonous •Interpretable decision making due to linear projection and additional insights into feature relevance	•Optimal performance only if groups form equal Gauss-distributed clusters •Linear projection can only solve simple classification tasks.

same time allow reasonable computing times for training (~1 h for RFESVM, training-time depends on FS-problem to solve) [34]. Advantages and disadvantages are summarized in table 8.

3.3. LDA and Mahalanobis classification

The classification algorithm used to evaluate the 15 feature subsets generated by all combinations of five feature extraction and three feature selection algorithms is linear discriminant analysis (LDA) followed by Mahalanobis distance classification. In case of g groups LDA performs the linear projection of the feature space into a $g-1$ dimensional discriminating space that minimizes within group scattering and maximizes between group scattering [40]. In this discriminating space points are classified into that group to which their Mahalanobis distance is minimal [31]. The scheme of the full data evaluation is shown in figure 13. First each unsupervised feature extraction method is applied to each (sensor) signal available generating five feature pools, one for each method. From each feature pool the best feature subset is selected using the three supervised feature selection methods. Finally, the best combination of the 15 combinations of feature extraction and feature selection is chosen based on the lowest cross-validation error of the final classification to solve the problem at hand, i.e. condition monitoring with different fault states as target classes. Note that this cross-validation loop has to include not only the classification itself but also feature selection and even extraction to account for possible overfitting in supervised feature selection and to ensure statistical stability in feature extraction. Keep in mind that a second, nested cross-validation loop is employed for feature subset size determination. The computational cost of this nested cross-validation is not prohibitive because the LDA projection can be computed analytically and is therefore very efficient. Additionally, LDA provides two further advantages over more complex classifiers like artificial neural networks or (nonlinear) support vector machines. First, the low dimensional representation achieved by LDA is easily visualized by 2D scatter plots that provide additional insight

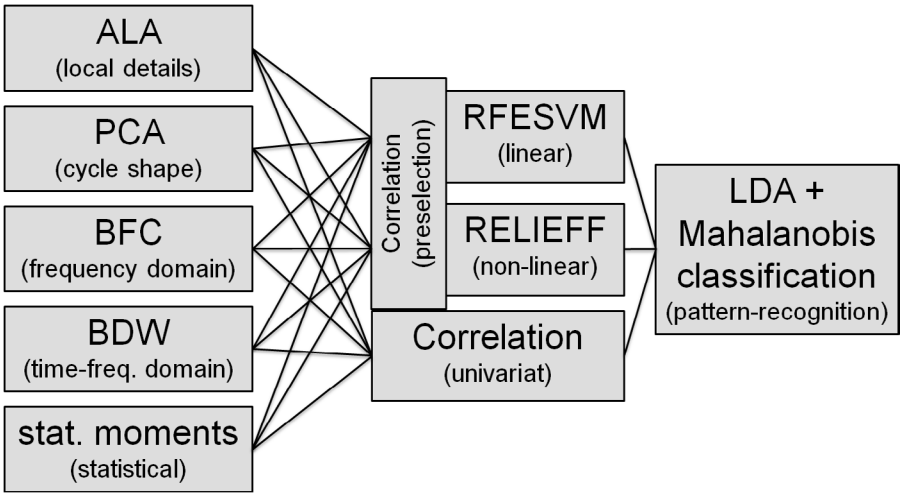


Figure 13. Schematic of the suggested algorithms for feature extraction (left), feature selection (center) and classification (right) and their combination for automated dimensionality reduction and classification. The performance of all 15 combinations of FE and FS methods are evaluated using LDA combined with Mahalanobis classification. The best combination based on the ten-fold cross-validation error is then chosen for the condition monitoring task at hand. © 2018 IEEE. Reprinted, with permission, from [25].

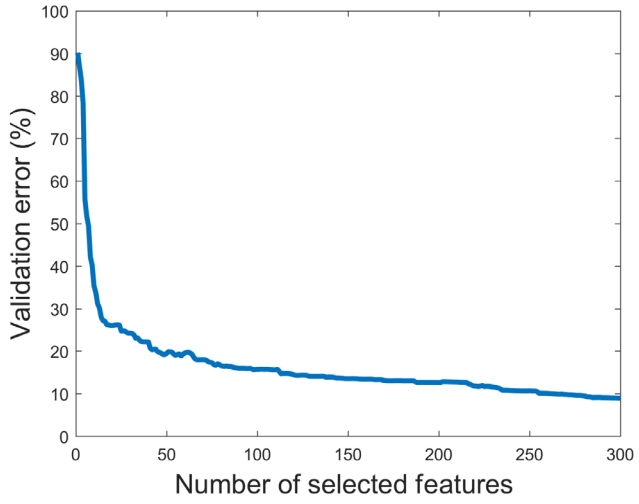


Figure 14. Ten-fold cross-validation error plotted over the number of features selected by their absolute Pearson correlation to the remaining lifetime. Features are added incrementally according to their ranking and for each feature subset the error is evaluated using ten-fold cross-validation. Small feature subsets contain insufficient information to predict the ECMs remaining lifetime, thus additional features greatly improve the classification performance up to 25% error for 20 features. Adding further features leads to only a small decrease in the prediction error. By adding even more features the classification error would rise again due to overfitting (not shown here).

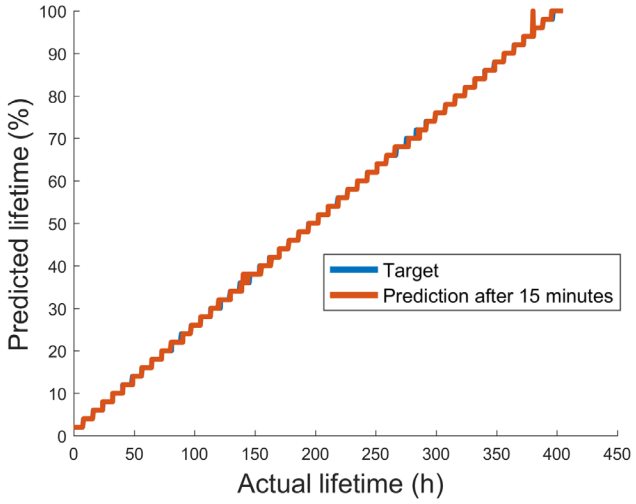


Figure 15. Ten-fold cross-validated lifetime prediction of an EMC averaged over 15 min. The prediction follows the actual lifetime of the EMC (blue) almost perfectly. Most prediction errors occur due to quantization at the class borders and are therefore negligible. Only a small section near the end of lifetime is severely misclassified.

into the data allowing intuitive understanding why a certain prediction is obtained for a given signal. Second, the linear nature of LDA allows analyzing the contribution of individual sensors, time segments or frequencies to the overall classification result. This information can be used to optimize the overall system, i.e. number and types of sensors used, sample rate and measurement time. Table 9 contains a summary of advantages and disadvantages.

3.4. Evaluation on multiple diverse datasets

To show the versatility of the explained approach and therefore its applicability in different classification scenarios it was tested on multiple datasets from very different domains. The evaluation datasets were intentionally chosen to be more diverse than the primary intended application condition monitoring. All applications have in common, that data can be treated as equal sized cycles that need to be classified. In [34] the approach was tested on eight different datasets and 17 different target values. Comparisons with the results previously achieved on these datasets show that on five out of eight tested datasets better results have been achieved compared to previously used approaches which were often specifically developed for the respective datasets. In the three other cases at least comparable results have been achieved and the approach failed on none of the tested datasets. These results were achieved without any manual parameter tuning and although a simple, linear classification algorithm was used. Furthermore, this set of algorithms is applicable both for smart sensors and

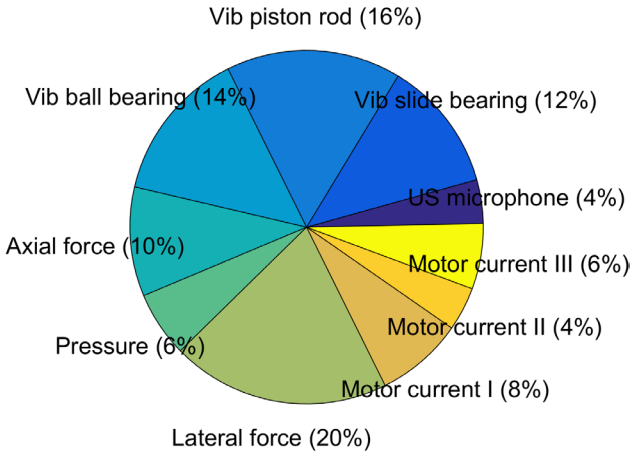


Figure 16. Best 50 features extracted by BFC and selected by Pearson correlation attributed to the underlying sensor. These features are selected first to predict the EMCs lifetime and therefore capture the most important information concerning the wear process. In fact, 28 of the best 30 features can be linked to physically relevant frequencies like harmonics of the motor speed. This information can be used to further optimize the measurement setup or to reduce the number of sensors used by eliminating the least relevant sensors.

big data, since all FE algorithms and feature preselection can easily be implemented on an FPGA and with the use of Map-Reduce allowing distributed and parallel computing.

To make full use of this modular and automated approach, data pre-processing and feature extraction need to be integrated in the sensor system to reduce the data load in the network and the cloud. However, this modular approach can also be used to design cost efficient sensor systems for smart monitoring applications. In this case, a complete 'over-instrumented' sensor set is used and the full sensor data are evaluated with the automated approach described above. The fairly simple and transparent algorithms allow identification of relevant sensors and features and, thus, the necessary acquisition bandwidth using an offline analysis. On this basis a greatly simplified sensor system can be defined for practical application. This approach would also allow choosing an application specific balance between sensor redundancy, i.e. to achieve robust operation, and cost efficiency.

4. Lifetime prediction of EMCs

To evaluate the fully automated approach in a big data condition monitoring scenario for predictive maintenance a test bench for lifetime tests of an EMC (Festo ESBF-BS-63-400-5P, ⌀ 63 mm, 400 mm stroke, 5 mm spindle pitch, max. axial load 7 kN) was set up to create a corresponding dataset. The EMC was operated at maximum velocity, repeatedly pushing and pulling against a load applied by a pneumatic cylinder, until failure of the EMC. This operation simulates quick wear-down of the EMC to record data over its complete lifetime. Data was recorded by three 1D vibration sensors sampled at 100 kHz (mounted on the engine side ball bearing, the end of the piston rod and the friction bearing), eight process sensors sampled at 10 kHz (axial force, torque, pneumatic pressure, velocity, piston position, electrical current, lateral force and vibration) and three motor current sensors sampled at 1 MHz. With this setup a total of 347.198 cycles (i.e. single push and pull) were recorded over a period of 21 d at a raw data rate of approx. 650 GB d^{-1}. Note that although condition monitoring of EMCs and the hydraulic application previously discussed are very different, the data structure of time series data can always be treated as working cycles (sliding windows in case of EMCs compared to actual working cycles for the hydraulic machine). Thus the basic data structure is identical and can therefore be treated using the same automated data processing approach. The diversity of the tested datasets thereby proves the versatility and flexibility of the chosen approach.

For machine learning the complete lifetime was split into 50 equally sized groups, i.e. representing 2% of the total lifetime each, and the automated approach described above was applied to identify differences between these groups. The validation error determined by ten-fold cross-validation shows that the most suitable combination of feature extraction and feature selection is the combination of statistical moments extracted on five equally sized cycle segments and their selection according to the largest absolute Pearson correlation to the target, i.e. the relative axis lifetime. As shown in figure 14 a minimum error of 9% is achieved when all 300 features are evaluated (15 sensors $*$ 5 segments $*$ 4 statistical moments). Note that the information about the EMC lifetime state is thereby determined for every single cycle, i.e. every 4.8 s. Since the system monitors the degradation of the EMC which is very slow compared to the cycle time this high information rate is unnecessary or even unwanted. Thus, further improvement is possible by averaging over multiple cycles. Figure 15 shows the results achieved by taking into account the predictions over the last 15 minutes. As shown the prediction follows the target vector very well. There is only one outlier (more than one class difference to target) at 96% lifetime and all other misclassifications occur at class boundaries where they are neither unexpected nor prohibitive to the application.

In addition, the approach allows to analyze which sensors are the most important for lifetime prediction. Figure 16 shows the sensors from which the top 50 features are extracted. This information can be used to further improve the sensor system and give further insight into the degeneration and decision making process.

5. Conclusion and outlook

This paper has shown for two relevant applications, a complex hydraulic machine and an EMC, how smart sensors combined with a well-chosen set of algorithms for machine learning can be used for condition monitoring to implement predictive maintenance. The suggested approach using complementary algorithms for feature extraction and selection automatically builds a validated data-based model to predict the learned typical component faults as well as remaining lifetime. Thereby the algorithms are also applicable to big data as shown in the EMC example. Additionally the linear character of the

classifier allows further insight into the decision-making process, e.g. importance of different sensors that can be used to optimize the overall system, so as to minimize the number of sensors used. The smart sensor is not only capable of monitoring the system condition but also the condition of the sensor network itself. This self-monitoring allows quantitative evaluation of simulated sensor faults. If a faulty sensor is detected the fault can be compensated by removing the affected sensor from the database and automatic re-training. As the suggested methods are complementary and were tested on several different datasets from different fields of application it should achieve comparable results in other condition monitoring scenarios.

The current (r)evolution of Industry 4.0 and industrial Internet of Things continues and is pushed—among other drivers—by the development of smart sensors. Note that only some of the capabilities of smart sensors have been addressed in this paper. One of the intended extensions is the use of unsupervised novelty detection to warn if the current sensor signal pattern is generated by an unknown fault scenario that does not fit any of the previously learned faults. In this way the new fault can be indicated early, although it cannot be identified yet. This is important since not all possible faults can be simulated during training. Another desired extension is online learning. Using online learning a previously unknown fault indicated by novelty detection and subsequently identified during maintenance can be included in the list of known faults to correctly identify the fault the next time it occurs.

ORCID iDs

Tizian Schneider https://orcid.org/0000-0003-3488-8944

References

[1] Stahl-Offergeld M 2011 Robuste dreidimensionale Hall-Sensoren für mehrachsige Positionsmesssysteme *Aktuelle Berichte aus der Mikrosystemtechnik—Recent Developments in MEMS, Band 20* (Aachen: Shaker-Verlag)

[2] Marien J and Schütze A 2009 Magnetic microsensors: Quo Vadis? *Proc. Sensor 2009* vol II pp 17–22

[3] Camps F, Harasse S and Monin A 2009 Numerical calibration for 3-axis accelerometers and magnetometers *Proc. 2009 IEEE Int. Conf. Electro/Information Technology, EIT 2009* pp 217–21

[4] Xie H and Fedder G K 2003 Fabrication, characterization, and analysis of a DRIE CMOS-MEMS gyroscope *IEEE Sens. J.* **3** 622–31

[5] Enoksson P, Stemme G and Stemme E 1997 A silicon resonant sensor structure for Coriolis mass-flow measurements *J. Microelectromech. Syst.* **6** 119–25

[6] Bur C, Bastuck M, Spetz A L, Andersson M and Schütze A 2014 Selectivity enhancement of SiC-FET gas sensors by combiningtemperature and gate bias cycled operation using multivariatestatistics *Sens. Actuators* B **193** 931–40

[7] Ochs T, Diehl L, Lehle W, Kern C, Stanglmeier F and Handler T 2013 Selbstüberwachung und online Verifizierung von Sensordaten im Kraftfahrzeug *11. Dresdner Sensor-Symp. 2013* pp 14–6

[8] Schmitt R H and Voigtmann C 2018 Sensor information as a service—component of networked production *J. Sens. Sens. Syst.* **7** 389–402

[9] Cachay J and Abele E 2012 Developing competencies for continuous improvement processes on the shop floor through learning factories—conceptual design and empirical validation *Proc. CIRP* **3** 638–43

[10] W3C Semantic Sensor Network Incubator Group 2005 Semantic Sensor Network Ontology www.w3.org/2005/Incubator/ssn/ssnx/ssn (Accessed: 28 January 2018)

[11] Schütze A, Helwig N and Schneider T 2018 Sensors 4.0—smart sensors and measurement technology enable Industry 4.0 *J. Sens. Sens. Syst.* **7** 359–71

[12] Helwig N, Pignanelli E and Schutze A 2015 Condition monitoring of a complex hydraulic system using multivariate statistics *2015 IEEE Int. Instrumentation and Measurement Technology Conf. (I2MTC) Proc.* pp 210–5

[13] Duda R O, Hart P E and Stork D G 2000 *Pattern Classiflcation* 2nd edn (New York: Wiley)

[14] Helwig N 2014 Intelligentes condition monitoring von hydraulischen Anlagen *AHMT 2014—Symp. des Arbeitskreises der Hochschullehrer für Messtechnik* pp 121–8

[15] Helwig N, Klein S and Schütze A 2015 Identification and quantification of hydraulic system faults based on multivariate statistics using spectral vibration features *Proc. Eng.* **120** 1225–8

[16] Helwig N, Pignanelli E and Schütze A 2015 Detecting and compensating sensor faults in a hydraulic condition monitoring system *Proc. Sensor* pp 641–6

[17] Doms M and Slatter R 2014 Magnetoresistive sensors for angle, position, and electrical current measurement in demanding environments *Proc. SPIE* **9113** 91130M

[18] Helwig N, Merten P, Schneider T and Schütze A 2017 Integrated sensor system for condition monitoring of electromechanical cylinders *Proceedings* **1** 626

[19] Schütze A 2015 MoSeS-Pro: Modulare Sensorsysteme für Echtzeit-Prozesssteuerung und smarte Zustandsbewertung für die Industrie 4.0, BMBF project funded in the funding area ‘Sensorbasierte Elektroniksysteme für Anwendungen für Industrie 4.0 (SElekt I4.0)

[20] Helwig N, Schneider T and Schütze A 2017 MoSeS-Pro: modular sensor systems for real time process control and smart condition monitoring using XMR-technology *Proc. 14th Symp. Magnetoresistive Sensors and Magnetic Systems* pp 15–22

[21] Jing C and Hou J 2015 SVM and PCA based fault classification approaches for complicated industrial process *Neurocomputing* **167** 636–42

[22] Shao R, Hu W, Wang Y and Qi X 2014 The fault feature extraction and classification of gear using principal component analysis and kernel principal component analysis based on the wavelet packet transform *Measurement* **54** 118–32

[23] Li B, Chow M-Y, Tipsuwan Y and Hung J C 2000 Neural-network-based motor rolling bearing fault diagnosis *IEEE Trans. Ind. Electron.* **47** 1060–9

[24] Paya B A, Esat I I and Badi M N M 1997 Artificial neural network based fault diagnostics of rotating machinery using wavelet transforms as a preprocessor *Mech. Syst. Signal Process.* **11** 751–65

[25] Schneider T, Helwig N and Schütze A 2018 Automatic feature extraction and selection for condition monitoring and related datasets *Proc. I2MTC 2018 (Houston, TX, USA)*

[26] Olszewski R T 2001 Generalized feature extraction for structural pattern recognition in time-series data *PhD Thesis* Carnegie Mellon University

[27] Anstey J, Peters D and Dawson C 2007 An improved feature extraction technique for high volume time series data *Proc. of the 4th Conf. on IASTED* pp 74–81

[28] Chakrabarti K, Keogh E, Mehrotra S and Pazzani M 2002 Locally adaptive dimensionality reduction for indexing large time series databases *ACM Trans. Database Syst.* **27** 188–228

[29] Keogh E J and Pazzani M J 1998 An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback *KDD'98 Proc. 4th Int. Conf. on Knowledge Discovery and Data Mining* vol 98 pp 239–43
[30] Wold S, Esbensen K H and Geladi P 1987 Principal component analysis *Chemometr. Intell. Lab. Syst.* **2** 37–52
[31] Schneider T 2015 Methoden der automatisierten Merkmalextraktion und -selektion von Sensorsignalen *Master Thesis* Saarland University
[32] Morchen F 2003 Time series feature extraction for data mining using DWT and DFT *Technical Report* No.33 (Dep. Mathematics Comput. Sci. Philipps-University Marburg) pp 1–31
[33] Guyon I 2003 An introduction to variable and feature selection *J. Mach. Learn. Res.* **3** 1157–82
[34] Schneider T, Helwig N and Schütze A 2018 Automatic feature extraction and selection for condition monitoring and related datasers *IEEE Int. Instrumentation & Measurement Technology Conf. (I2MTC)*
[35] Guyon I, Gunn S, Nikravesh M and Zadeh L A 2006 *Feature Extraction—Foundations and Applications* (Berlin: Springer) (https://doi.org/10.1007/978-3-540-35488-8)
[36] Abe S 2010 *Support Vector Machines for Pattern Classification* 2nd edn (London: Springer) (https://doi.org/10.1007/978-1-84996-098-4)
[37] Kononenko I and Hong S J 1997 Attribute selection for modelling *Future Gener. Comput. Syst.* **13** 181–95
[38] Kohavi R and John G H 1997 Wrappers for feature subset selection *Artif. Intell.* **97** 273–324
[39] Wettschereck D, Aha D W and Mohri T 1997 A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms *Lazy Learning* (Dordrecht: Springer) pp 273–314
[40] Duda R O, Hart P E and Stork D G 2001 *Pattern Classification* 2nd edn (New York: Wiley)

5.4 Paper 3: Machine Learning in Industrial Measurement Technology for Detection of Known and Unknown Faults of Equipment and Sensors

As described in the previous paper, the automated machine learning toolbox is limited to detecting known faults that have been sufficiently sampled. However, acquiring enough data samples with every possible fault is impossible or at least uneconomical in many application cases. In those cases, novelty detection is a good approach. However, as shown in the following paper, novelty detection is suitable not only for unknown or underrepresented fault detection [115] but also for outlier detection [116] and monitoring the validity of supervised machine learning decisions.

For these applications, novelty detection defines measures of novelty and places thresholds on these measures to determine whether a given sample is novel. The novelty scores used mainly differ in their approach to defining the novelty measure [117]. An overview of the different typical approaches can be found in Section 2 of the following paper.

The following paper presents novelty detection in the context of the automated machine learning toolbox machine learning pipeline with feature extraction, selection, classification/regression, and the usage of the respective algorithms. Concerning novelty detection in general and especially in this context, some remarks have to be made:

Feature standardization: Most novelty scores are sensitive to feature scaling or need special precautions to become invariant to feature scaling. Therefore, feature standardization is advised before novelty detection.

Choosing a novelty measure: Many authors recommend receiver operating characteristic (ROC-) curves, especially the area under the curve (AUC), to compare different scores. As the AUC is a threshold-independent measure of contrast [0,1] between normal and novel data provided by the respective measure, this method sounds appealing [118]. However, this measure is unreliable for the detection of previously unknown faults. First, it requires samples of an unknown fault that are likely unavailable. Second, even if samples of a previously unseen fault are available, the comparison by AUC will be specific to this fault as a testing error in supervised learning is to a specific test set. It does not guarantee that it is transferable to another fault that might be better detectable by a different metric.

Choosing a suitable threshold: As choosing a suitable threshold is highly dependent on the respective application, most literature refers to this fact and limits itself to suggesting new novelty measures. Other papers suggest general, application-independent heuristics [119] or elbow criteria on the ROC curve [118]. As the threshold highly depends on the use-case and ROC curves suffer from the abovementioned issues, the following paper also suggests methods of use-case-specific threshold setting for each application.

Validation: Even though novelty detection is considered unsupervised or semi-supervised, k-fold or group-based cross-validation is needed to confirm the statistical validity of the model of normality built by the respective algorithm. A simple example would be a novelty score based on the 1-Nearest-Neighbour distance that is exactly zero for all training points since the point would be contained in the training set. Therefore, a proper estimate of AUC or other metrics could only be given for samples outside the training set.

Machine Learning in Industrial Measurement Technology for Detection of Known and Unknown Faults of Equipment and Sensors

Tizian Schneider[1], Klein Steffen[1], and Andreas Schütze[1,2]

[1]Saarland University, Lab for Measurement Technology, Saarbrücken, Germany
[2]Centre for Mechatronics and Automation Technology (ZeMA gGmbH), Saarbruecken, Germany

tm – Technisches Messen (2019), 86 (11), 706-718

The original paper can be found online at https://doi.org/10.1515/teme-2019-0086.

© Used with permission of Walter de Gruyter and Company, from Machine Learning in Industrial Measurement Technology for Detection of Known and Unknown Faults of Equipment and Sensors, Schneider, Tizian; Klein, Steffen; Schütze, Andreas, 86, 11, 2019; permission conveyed through Copyright Clearance Center, Inc.

Tizian Schneider*, Steffen Klein, and Andreas Schütze

Machine learning in industrial measurement technology for detection of known and unknown faults of equipment and sensors

Machine Learning in der industriellen Messtechnik zur Erkennung bekannter und unbekannter Anlagen- und Sensorfehler

https://doi.org/10.1515/teme-2019-0086
Received June 11, 2019; accepted August 12, 2019

Abstract: This paper focuses on the application of novelty detection in combination with supervised fault classification for industrial condition monitoring. Its goal is to provide a guideline for engineers on how to apply novelty detection for outlier detection, monitoring of supervised classification and detection of unknown faults without the need of a data scientist. All guidelines are demonstrated by means of a publicly available condition monitoring dataset. In each application case the results achieved with different common novelty detection algorithms are compared, advantages and disadvantages of the respective algorithms are shown. To increase applicability of the suggested approach visualization of results is emphasized and all algorithms have been included in a publicly available data analysis software toolbox with graphical user interface.

Keywords: Machine learning, novelty detection, condition monitoring.

Zusammenfassung: Dieser Aufsatz befasst sich mit der Anwendung von Anomaliedetektion in Kombination mit überwachter Schadensklassifikation in der industriellen Messtechnik. Ziel ist es Ingenieuren einen Leitfaden an die Hand zu geben, wie Anomaliedetektion auch ohne Data Scientist zur Erkennung von Ausreißern, zur Kontrolle der überwachten Schadenserkennung und zur Erkennung bisher unbekannter Maschinenstörungen eingesetzt werden kann. Alle empfohlenen Vorgehensweisen werden an einem öffentlich zugänglichen Datensatz zum Thema Zustandsüberwachung demonstriert. In jedem Anwendungsszenario werden die mit unterschiedlichen und weit verbreiteten erreichten Algorithmen zur Anomaliedetektion verglichen und Vor- und Nachteile aufgezeigt. Um die Hemmschwelle beim Einsatz der vorgeschlagenen Herangehensweise zu senken wird großer Wert auf Visualisierungen von Ergebnissen gelegt. Weiterhin sind alle verwendeten Algorithmen Teil einer kostenlosen Software zur Datenanalyse mit grafischem Benutzerinterface.

Schlagwörter: Maschinelles Lernen, Anomaliedetektion, Zustandsüberwachung.

1 Introduction

Predictive maintenance and continuous monitoring of product quality are central promises of Industry 4.0 and the Industrial Internet of Things (IIoT) [1]. Machine learning (ML) has been shown to help fulfil these promises by learning fault-patterns in sensor data and deriving relevant information that cannot be measured directly, like wear progression, remaining useful lifetime, expected product quality before end-of-line testing, broken machine parts, sensor interference and many others, from symptoms [2]. This potential of data based modeling has been shown in multiple research papers [3] and is still an ongoing research subject. However, the usual approach of learning fault patterns from sensor data, i. e. supervised learning, requires data from a faulty machine and is therefore limited to the recognition of known faults. To detect unknown faults and faults that have not occurred yet an unsupervised learning approach is needed. As in the field of supervised learning a huge variety of available algorithms exists for novelty detection [4] and it is not trivial to select the optimal one or at least a suitable one for the task at hand. A study of 340 papers proposing new algorithms for data mining by Keogh and Kasetty showed that in these papers the algorithms were tested on average on 1.3 different datasets and compared to an average of only 0.9 similar algorithms [5]. This demonstrates that

***Corresponding author: Tizian Schneider,** Zentrum für Mechatronik und Automatisierungstechnologie ZeMA, Aktorik und Sensorik, Eschberger Weg 46, Gewerbepark, Gebäude 9, 66121 Saarbrücken, Saarland, Germany, e-mail: t.schneider@zema.de
Steffen Klein, Andreas Schütze, Universität des Saarlandes, Lehrstuhl für Messtechnik, 66123 Saarbrücken, Saarland, Germany

in the ML community, the emphasis is mainly on new algorithms and less on the selection of suitable algorithms for a wide range of applications. In addition, many algorithms require complex tuning of hyperparameters to adapt to datasets. Therefore, algorithm selection and hyperparameter tuning require data scientists which increases the cost of ML projects and slows down widespread use of the technology by engineers in industry.

This paper studies simple application rules for novelty detection based on automated algorithms and simple heuristics to enable engineers to employ ML in their application cases without the need of data scientists and therefore aims to lower the barrier for using ML for quality monitoring and fault detection. To do this the paper shows how novelty detection can be employed for outlier detection, monitoring of supervised classification and detection of unknown faults. In this paper several typical faults of hydraulic machines, namely cooling power degradation, valve switching characteristics degradation, accumulator pre-pressure loss and internal pump leakage, are considered. In each application scenario guidelines for the application of novelty detection are derived and the results of different approaches to novelty detection in combination with feature extraction and threshold estimation are compared on a condition monitoring dataset to allow choosing a suitable algorithm. To increase interpretability of the results visualization is emphasized. For applicability and user acceptance all algorithms and visualizations are publicly available as part of a data analysis toolbox with graphical user interface [6].

2 Algorithms and datasets

This paper is based on a supervised classification framework for industrial condition monitoring [2] following the typical steps of feature extraction, dimensionality reduction and classification of faults. This framework can extract information from time, frequency and time-frequency domain by automatically applying multiple, complementary algorithms and choosing the best combination for the respective classification task. In this study the framework is complemented and expanded with novelty detection algorithms for outlier detection, justification of classification results and for detection of unknown faults and anomalies.

In all application scenarios a novelty detection algorithm builds a model of normality based on training data and scores new measurements depending on their novelty or similarity compared to the training data. Consequently, novelty detection is defining a measure for normality of sensor data (patterns) and a method of setting a threshold on this measure to decide whether new data is novel or normal. Different algorithms for novelty detection can be categorized by their approach to define the measure for normality [4] however other categorizations exist [7, 8]. The most frequently used approaches following the categorization of Pimentel et al. [4] are:

Probabilistic: Probabilistic approaches estimate the probability density function (PDF) D of the training data [4]. The estimate D' of the PDF is used as similarity measure for novelty detection. For new data points the value of D' will be high if it is drawn from the same distribution as the training data, i. e. if it is normal. A well-known algorithm implementing this approach is the Gaussian Mixture Model (GMM) [9] which models the PDF as a superposition of multiple Gaussian distributions using Expectation Maximization [7]. The number of Gaussian components in the model can be estimated using heuristics such as the Bayesian Information Criterion [10] which is given by $[-2\log(L) + \log(n)d]$ where L is the likelihood function and d is the number of parameters to be estimated.

Distance based: Distance based novelty detection methods use distances (often the Euclidean distance) between single data points as novelty measure [11]. If new data points lie close to the training data, i. e. if they are normal, their novelty score will be low. The most commonly used algorithm of this class is K Nearest Neighbors (KNN). To balance adaption to the training data, smoothness of the decision border and performance while avoiding parameter tuning K is set to 5, i. e. the novelty score is the total distance to the five nearest neighbors. A study on how the parameter K affects the novelty score can be found in [12].

Domain based: Domain based algorithms model the boundary between novel and normal data points [13]. Support Vector Machines (SVM) are widely used techniques for forming decision boundaries separating data of different classes. For novelty detection One Class SVM are used to separate the training data from the origin of the coordinate system. Choosing a radial kernel like the Gaussian kernel is equivalent to forming a convex shell around the training data. The distance to this shell is used as similarity measure [14]. This also demonstrates the mathematical similarity between distance based and domain based approaches, since SVM use the dot or scalar product between two training data points as central similarity measure. However, in domain based approaches this similarity measure is not used directly to classify normal and novel data points but is used to derive a hyperplane separating normal and novel data.

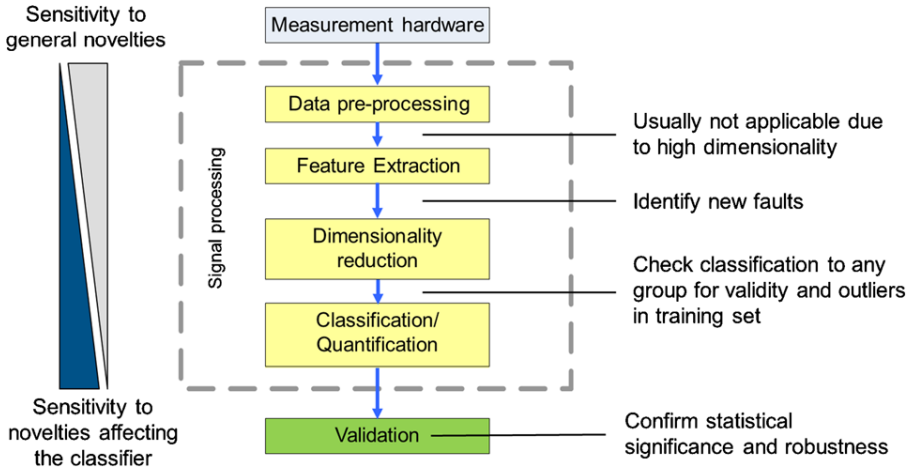


Figure 1: Processing steps for supervised classification of known faults with potential of and restrictions for novelty detection applied between these steps. The later novelty detection is applied the stronger the training data is already adapted to the original supervised classification task. Thus, the data processing steps increase the sensitivity of novelty detection to novelties affecting the classifier and decrease the sensitivity to general novelties that are suppressed during data processing.

Reconstruction based: Reconstruction based algorithms learn a low-dimensional representation of the training data that, as in feature extraction, captures characteristics of the data and allows reconstruction with minimal loss of information. If novel data do not fit the learned patterns it will be reconstructed with a high approximation error, which can therefore be used as novelty measure. Autoencoders (AEC) using neural networks [15] are most widely used in this field. The network used in this study is composed of a single hidden layer with sigmoid activation function. The number of neurons in this layer being chosen by the heuristics in [16] is the square root of the number of input neurons. Note that more recent approaches with deep autoencoders are more promising concerning performance. However these approaches require several parameters to be tuned (e. g. the number of neurons in each layer) and are therefore not suited for the intended application by engineers.

This paper shows exemplary results achieved with GMM, KNN, SVM, and AEC to represent the four approaches. All algorithms have been integrated into the Data Analysis, Visualization, Verification and Validation Environment (DAV^{3}E) toolbox implemented in MATLAB which is available as open source [6].

For each of these methods the unavoidable tradeoff between number of false positives and number of false negatives is decided by the threshold set on the respective novelty or similarity score. A high threshold will allow a low false positive rate (most normal samples will be assigned to the normal class) but will lead to a high false negative rate (many novelties remain unnoticed). On the other hand, a low threshold will lead to a high false positive rate (many false alarms). For the most common or popular algorithms there are heuristics to set the threshold. However, the applicability of these heuristics strongly depends on the specific dataset and the application scenario. Therefore, this paper describes an approach for setting a suitable threshold in every application scenario.

Direct application of novelty detection on raw data is usually not possible due to overfitting and the "curse of dimensionality" [17] that prevents assessing similarities in high dimensional spaces due to small, random variations adding up to significant differences which are then interpreted as novelties. At the same time the tradeoff between sensitivity to new faults and sensitivity to novelties that disturb classification of known faults depends on the data processing step after which novelty detection is applied. The typical data processing steps of supervised fault classification are shown in Figure 1. Every step is designed to reduce the amount, i. e. dimensionality, of data by concentrating the contained information in as few features or variables as possible, which is usually achieved by removing sensor characteristics and information not contributing to the classification of known faults. Therefore, novelties that do not affect the classification of known faults are suppressed with increasing dimensionality reduction. On the other hand, small novelties that are highly relevant for fault classification might get masked by other variations in the earlier stages of data processing.

To demonstrate the application of novelty detection a publicly available dataset for condition monitoring is used. The dataset was generated from a complex hydraulic

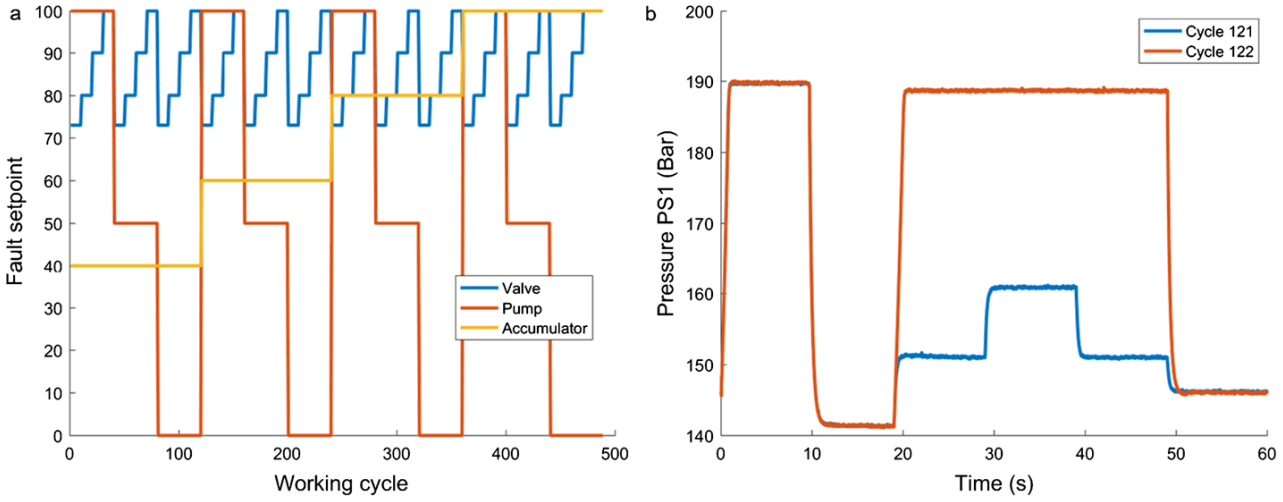


Figure 2: Design of experiment of the hydraulic condition monitoring dataset used for demonstration of the condition monitoring approach. During the experiment all possible combinations of faults (cooling power degradation, valve switching degradation, pump leakage, and accumulator pressure loss) are simulated in multiple grades of severity. (a): One section of the complete experiment showing the variation of the simulated valve, pump and accumulator fault setpoints. This is repeated for the cooler running at 100 % (normal operation), 20 % and 3 % duty cycle. (b): Pressure measured at the main valve during the machine working cycle which is repeated during the experiment. Cycle 121 represents a normal cycle, cycle 122 an anomaly, i. e. valve switching failure occurring at high temperatures (cooler running at 3 % duty cycle).

machine that is capable of simulating four fault scenarios that are typical for hydraulic machines [18]. The set-up simulates (a) cooling power decrease by reducing the fan duty cycle, (b) degradation of valve switching characteristics by reducing the control current, (c) internal pump leakage by switchable bypass orifices and (d) gas leakage of the accumulator by switching between accumulators at different reduced precharge pressures. The dataset includes data from experiments simulating the combination of all possible faults in multiple grades of severity. The design of experiment for faults b-d is shown in Figure 2 (a); this is repeated three times for cooler fan duty cycles (DC) of 100 % (normal operation), 20 % and 3 %, respectively. During the experiment data is recorded by a total of 23 sensors for pressure, temperature, electrical power, vibrations and flow rate, while the machine performs a constant working cycle (cycle 121) shown in Figure 2 (b) as measured by a pressure sensor (PS1). Cycle 122 in Figure 2 (b) shows a randomly occurring anomaly of the main valve, which does not switch correctly in approx. 20 % of the cycles at high oil temperature (cooler running at 3 % DC). This random anomaly is one exemplary target for a novelty detection. The dataset is available for download at the UCI ML repository [19]. A more detailed description of the set-up can be found in [18].

For novelty threshold tuning two different types of validation are employed. The first one is the widely used 10-fold cross validation, in which the training dataset is randomly split into ten parts. Nine of these partitions are used as training data while the tenth part is used for testing. The training is repeated ten times with a different testing set in each iteration and the results on all test sets are reported as final estimation. Cross validation will reveal overfitting and statistical variations in model applications. The second validation technique is group-based cross validation, in which the training and test data are not partitioned randomly but based on the group affiliation of the cycles. In the dataset described above the groups are given by the different grades of severity for the different faults. As in 10-fold cross validation training is performed on all but one group which is used as test data. Reporting the results of each test group provides insight into systematic variations in model applications and checks the model's ability to interpolate between groups.

3 Novelty detection of outliers

The most common application of novelty detection is outlier detection. The goal is to gain insight into the data quality and to find outliers, i. e. data samples that differ significantly from the regular data distribution, and remove them before supervised modeling to increase the classification performance and to emphasize the regular data distribution. To achieve this goal only features previously

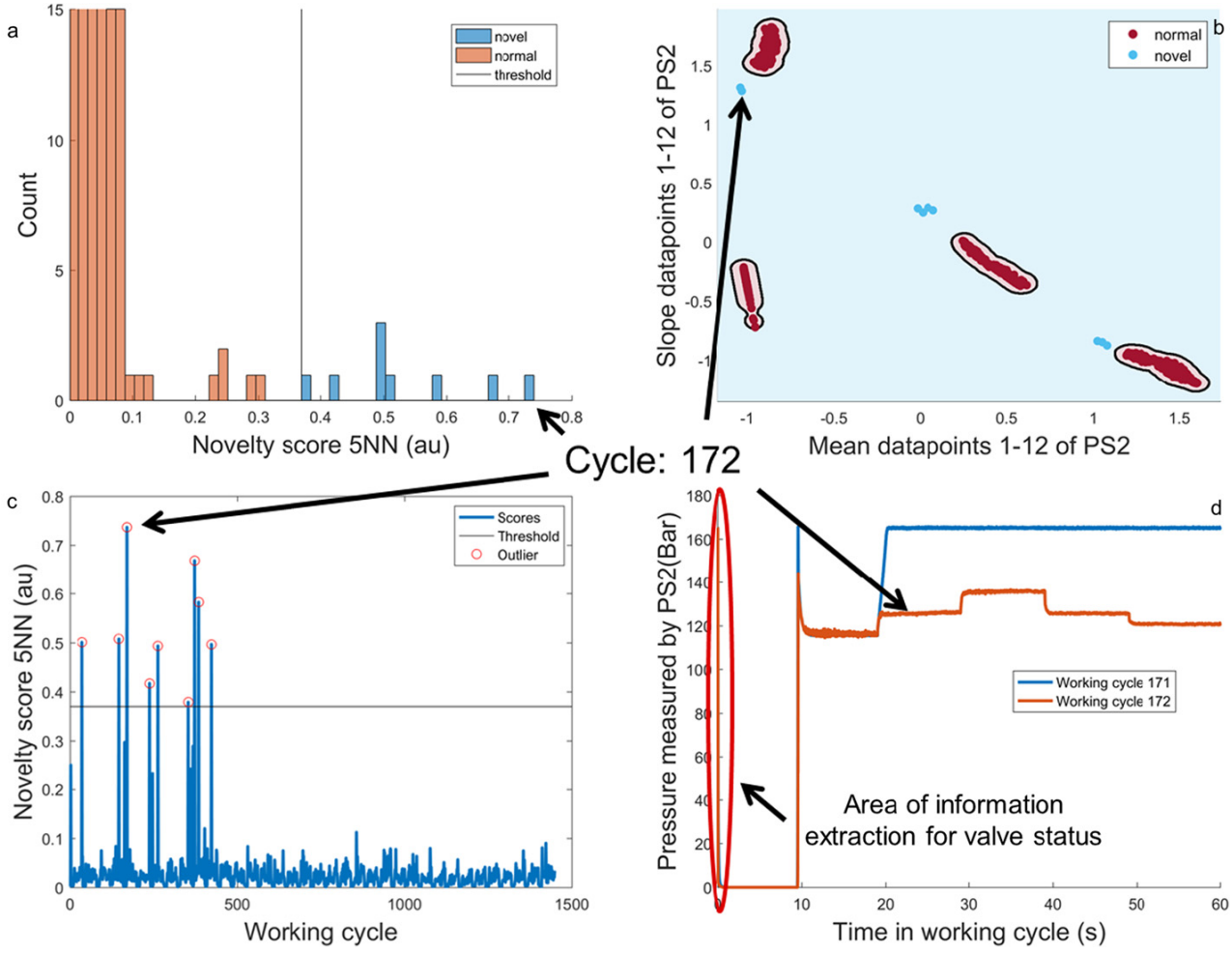


Figure 3: Novelty detection results achieved with five Nearest Neighbors (5NN) algorithm shown as histogram plot (a), territorial plot (b) or progression plot (c). The histogram plot shows the distribution of the novelty scores with outliers being clearly visible as single samples distinct from the majority of data samples. The territorial plot shows the selected threshold, i. e. decision boundary of 5NN in feature space. The progression plot shows the novelty score assigned to every working cycle during the experiment (cf. Figure 2 (a)). (d): Working cycle with highest novelty score (cycle 172) and the preceding cycle causing the novelty (pressure drop from 170 bar instead of 120 bar) measured by pressure sensor two located behind the main valve. The area of information extraction used by the supervised learning approach for classification of valve switching degradation is indicated.

selected for modeling are considered for outlier detection since they expose even small variations that might affect the classifier which is based on these features (compare Figure 1). The following approach is applied to identify outliers:

1. Extract and select features for the desired fault classification task using supervised learning with the automated algorithm toolbox described in [2].
2. Standardize features used for modeling to compensate scaling effects, i. e. to prevent large scale features from dominating the novelty score due to their large variance. For example, the KNN distance score might solely depend on one feature, if the scale of this feature is several orders of magnitude larger than that of all other features. Standardization weighs all features equally for unsupervised learning.
3. Train the novelty detection algorithm on the standardized features.
4. Use the histogram plot (compare Figure 3 (a)) to identify outliers as data samples with high novelty score or low similarity score, respectively.
5. Select a suitable novelty threshold based on the robustness of the classifier used for fault classification. The more robust the classifier the fewer outliers need to be removed.
6. Use progression plots (compare Figure 3 (c)) showing the predicted novelty score over the progression of the data acquired during the experiment to correlate out-

liers to physical events. Use physical interpretation to readjust the novelty threshold to exclude actual physical outliers.

Figure 3 shows the results of the described approach using KNN novelty detection on the data used for classification of valve degradation. The best classification results are achieved using Adaptive Linear Approximation (ALA) for feature extraction and Recursive Feature Elimination Support Vector Machines (RFESVM) for feature selection. ALA segments the machine's working cycle into multiple linear segments. RFESVM recursively eliminates features from the feature pool which contribute least to the data separation achieved with a linear SVM. The most relevant features selected for the classification of valve switching degradation are the mean value and the slope of pressure sensor two (PS2) during the first 0.12 seconds of each working cycle. In these 0.12 seconds the main valve is shut completely (pressure at PS2 drops to 0 bar). The resulting data distribution of the standardized mean and slope values is shown in the territorial plot in Figure 3 (b) which – based on 2D data – provides an intuitive insight into the shape of the decision boundary of the chosen novelty detection algorithm. Each of the four clusters represents one simulated grade of severity for valve switching degradation. The threshold for the decision boundary has been chosen using the histogram in Figure 3 (a) to exclude severe outliers. Simultaneously, Figure 3 (a) shows that outliers can be identified with a high contrast. Physical interpretation of the outliers is possible using the progression plot in Figure 3 (c). As shown, all outliers occur within the first third of the characterization measurement (cooler working at 3 %). Since the framework used for feature extraction and selection allows simple trace back of the features to the original raw data, Figure 3 (d) can be created to provide additional insight. In fact, the outlier is caused by a valve switching failure in the cycle before the outlier itself, thus the pressure drops to 0 bar from 170 bar instead of the usual 120 bar. This emphasizes the importance of physical interpretability since it shows that both the outlier cycle itself (affected by the fault) as well as the cycle before (faulty cycle) should be removed before modeling. A purely data driven approach would only remove the affected cycle and ignore the root cause. Note that outliers like cycle 122 shown in Figure 2 (b) are not evident in Figure 4 because their waveform does not differ from a normal cycle in the area where information is extracted for valve switching degradation (Figure 3 (d)). Therefore they do not affect the desired classification of valve switching degradation and are consistently suppressed by the employed outlier detection.

For an intuitive view on the different approaches to novelty detection Figure 4 (a), (b) and (c) show the territorial plots of AEC, GMM and SVM, respectively. For all three algorithms the dense clusters lie within the normal section of the data resulting in low novelty scores. However, only GMM (b) and SVM (c) form tight boundaries around these dense clusters whereas AEC tries to form a single cluster resulting in low novelty scores for most outliers. For comparison of the different approaches Figure 4 (d) shows the standardized novelty scores for KNN and AEC and the negative standardized similarity scores for GMM and SVM. KNN shows the highest peaks and therefore the strongest contrast between normal data points and outliers and should therefore be preferred for outlier detection. The SVM similarity scores saturate for outliers outside the SVM margin at –1 and, thus, result in equally high peaks in Figure 4 (d). The similarity score of GMM decreases exponentially with the distance to the cluster center resulting in barely visible contrast for outliers on the depicted linear scale making threshold tuning difficult. Finally, AEC fails to capture the individual clusters resulting in practically no contrast between normal data and outliers. This is a result of the simple structure of the used autoencoder with only one hidden layer. As recent progression in the field of deep autoencoders suggest, autoencoders can learn such complex structures. However, this would require hyperparameter tuning and therefore is unsuitable for the intended application by engineers. In conclusion, KNN should be the preferred algorithm for outlier detection in this case. In general, the histogram plots of multiple novelty detection algorithms should be evaluated in combination with raw data plots (e. g. Figure 3 (c)) to decide which data should be treated as outliers.

Table 1 shows the number of outliers detected for the different algorithms tested. Note that the number of outliers depends on the selected novelty threshold. For example KNN also detects 13 outliers when the threshold is set to 0.2 (cf. Figure 3 (a)). It is therefore recommended to use raw data plots (cf. Figure 3 (d)) to decide which data should be treated as outliers and which should not.

4 Monitoring supervised learning

A second application for novelty detection is monitoring of supervised fault classification, i. e. checking whether a new data sample can be assigned to any of the known fault groups. If the process generating the data to be classified changes due to new circumstances, the model trained on the original training data is no longer applicable and the

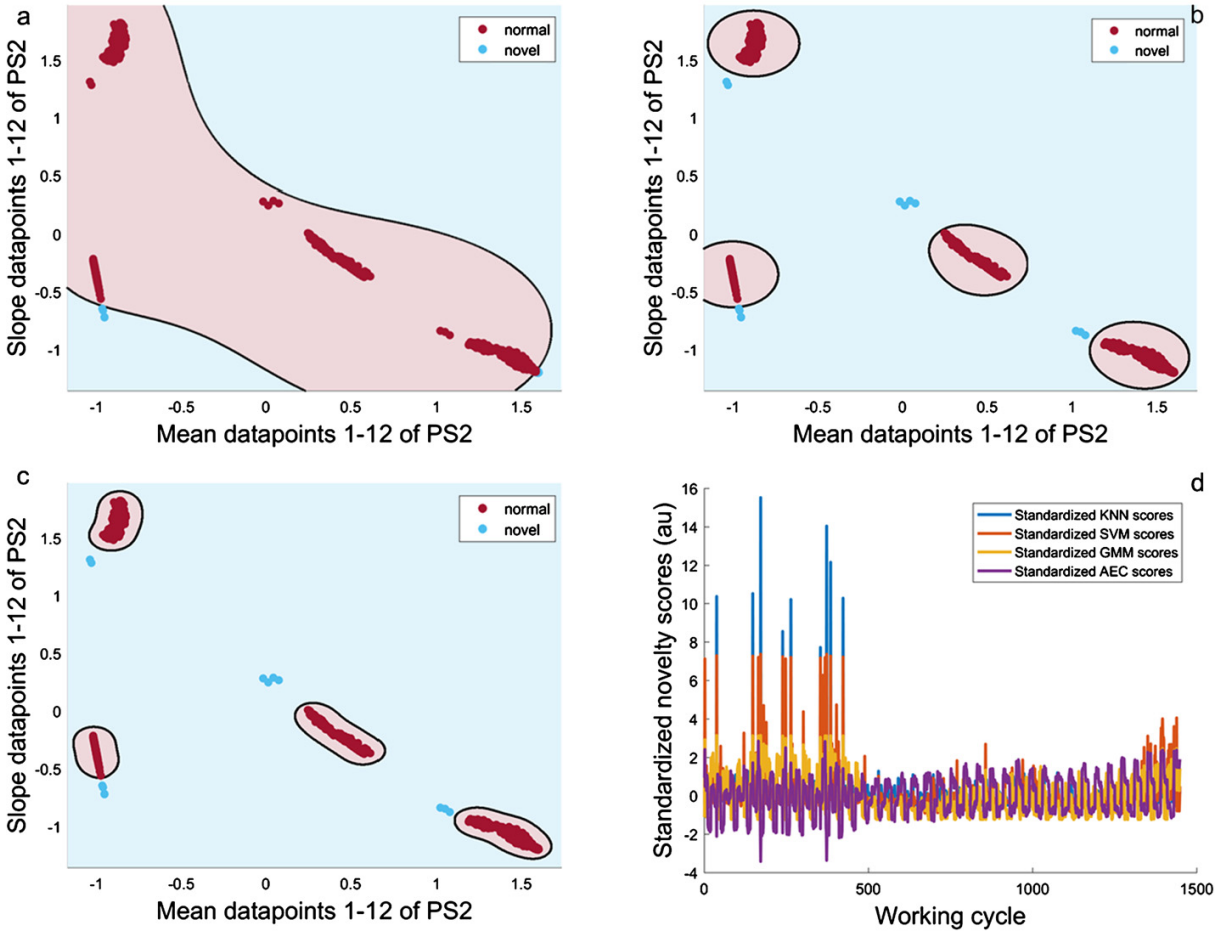


Figure 4: Territorial plots showing the novelty detection boundaries of AEC (a), GMM (b) and SVM (c), respectively, with data samples marked as normal (red) or novel (blue) by the respective algorithm. (d): Standardized novelty scores of KNN and AEC and negative standardized similarity scores of SVM and GMM as progression plot over the experiment. High peaks imply high contrast of novel points compared to the normal variation (standardized with mean = 0 and standard deviation = 1). The saturation of SVM similarity scores for outliers at –1 leads to peaks with equal height of the standardized novelty scores.

Table 1: Number of outliers detected by different novelty detection algorithms after the threshold was set according to step 5 in the above approach and in case of KNN lowered to exclude all 13 known outliers. Choosing the respective thresholds non false positives are detected. Lowering the threshold of AEC the algorithm would start to detect false positives.

Algorithm	KNN	AEC	GMM	SVM
Number of outliers	13/13	5/13	13/13	13/13

user needs to be notified to correct the problem. This application scenario therefore focuses on the detection of novelties that affect the classifier. Thus, the features used for novelty detection are the same as those used for the classification. For detection of unknown faults refer to Section 5. The main difficulty in this application is choosing a suitable threshold on the novelty score that allows the classifier to interpolate different severities of fault scenarios and at the same time still recognizes small changes in the process generating the data. The approach taken is as follows:

1. Extract and select features for the desired fault classification task using supervised learning with the automated algorithm toolbox described in [2].
2. Standardize features to be used for modeling to compensate for scaling effects, i. e. to prevent large scale features from dominating the novelty score due to their large variance.
3. Train a novelty detection algorithm on these standardized features.
4. Set a threshold using histogram plots and group-based validation, i. e. groups left out during model building are just recognized as normal to allow for interpolation between different fault states.

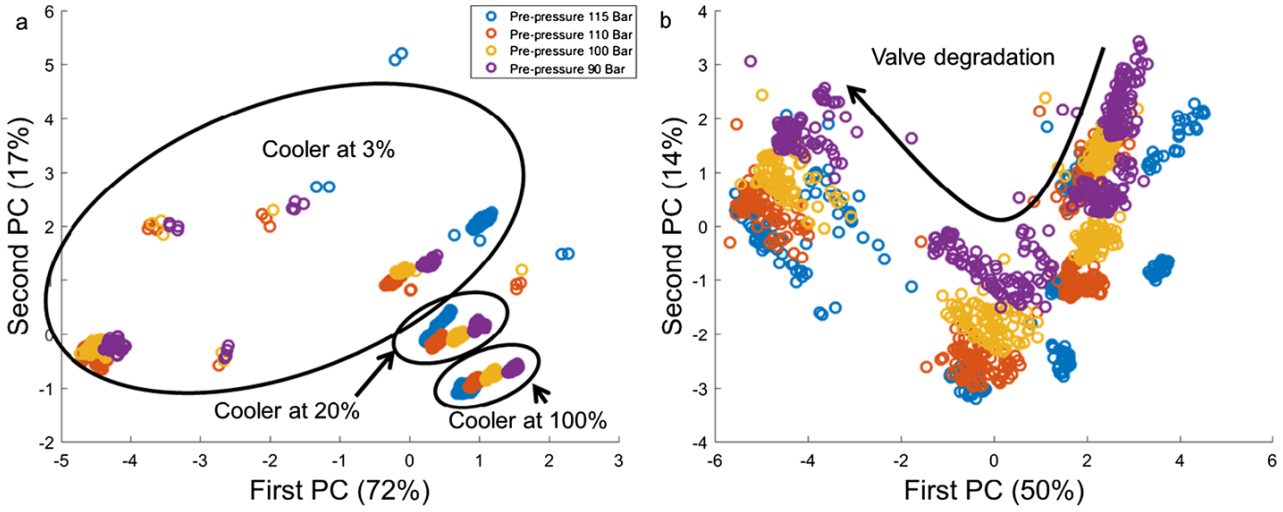


Figure 5: 2D Principal Component Analysis of features used for detection of accumulator pre-pressure loss when using only samples with cooler running at 100 % DC (a) or valve operating at 100 % (b). Despite having been excluded during training the influence of cooling power reduction (a) and valve switching degradation (b), respectively, are clearly visible in the data and can be detected as novelty.

For group-based validation the novelty detection algorithm is trained with all but one grade of fault severity and novelty scores are predicted for the left out group. This process is repeated for each group and the resulting novelty scores are shown as histogram. Choosing a threshold that just accepts few data samples as normal will then allow the classifier to interpolate between grades of fault severity. Leaving out one group during training increases, in linear models doubles, the necessary interpolation distance because the classifier has to interpolate from one group to the next but one while, when all groups are considered during training, it only needs to interpolate from one group to the group boundary of the adjacent group. In novelty detection this leads to selection of rather high threshold values. However, simply halving the threshold is not possible even for linear models due to the non-linearity of the novelty scores and does not guarantee better interpolation capability.

To show the results of this approach multiple process change scenarios are generated from the hydraulic dataset by ignoring the influence of one fault completely and testing how this affects supervised classification of the remaining faults. Since one of the intentions of this paper is to give rules on how to set suitable thresholds on novelty scores the only evaluation scores that compare the different novelty detection algorithms for that specific threshold are shown. Measures of comparison that are independent of the threshold (e. g. AUC of ROC-Plot) are not shown intentionally. There are some examples of classification tasks which are not affected by variations introduced by other faults. One of them is cooling power degradation which is always being recognized simply from a temperature sensor. Thus, even though other variations were not taken into account during training the cooler fault can always be classified correctly and every new data sample is therefore correctly identified as normal. The same holds for the detection of the valve switching degradation, where a correct classification rate of at least 99 % is achieved independently of all other faults. Both cooler and valve variations lead to substantial variations in the measurement data which are easily detected while pump and accumulator faults lead to more subtle changes in the data. This is shown in Figure 5 depicting the 2D Principal Component Analysis (PCA) of the features used for novelty detection in two different scenarios. In both cases features have been extracted and selected for classification of accumulator pre-pressure loss, which is shown by color, i. e. each step of pressure loss is shown in a different color. In Figure 5 (a) only cycles with the cooler running at 100 % DC have been used for training of the feature extraction and feature selection, i. e. the influence of cooling power degradation (cooler working at 20 % and 3 % DC, respectively) is ignored during training. Similarly, the influence of valve switching degradation is ignored in Figure 5 (b) and only cycles with the valve operating at 100 % are used for training the feature extraction and selection. All results for the analyzed application scenarios can be found in Table 2. The table first shows the true positive rate on the training data indicating how many training data points are correctly classified as normal. This is 100 % if none of the training data samples are mistaken for novel. Second, the table shows the percentage of correctly classified samples

Table 2: Novelty detection rates of different algorithms on different monitoring scenarios for supervised classification of the respective target fault when additional variations are added that were not included in the training. For example, the first row shows results for classification of valve switching characteristics when only data with the cooler operating at 100 % DC is used for training. The table shows the true positive rate on the training data in column 3, indicating how many training data points are correctly classified as normal (100 % if none of the training data samples are mistaken as novel). The 4th column shows the percentage of correctly classified samples which are detected as normal. These are the samples with novel variations that are still correctly classified by the supervised classifier and are identified as normal by the novelty detection (100 % indicates the desired case where none of the samples are identified as novel because the superimposed variations do not affect the supervised classification). The 5th column shows the percentage of samples misclassified by the supervised classifier due to superimposed variations that are correctly identified as novelty because the variations affect the supervised classifier (100 % indicates the desired case where all samples are identified as novel because the superimposed variation leads to false results of the supervised classification).

Classification target	Ignored variation	True positive rate on training data for KNN/SVM/GMM/AEC	Correctly classified samples detected as normal for KNN/SVM/GMM/AEC	Misclassified samples detected as novelty for KNN/SVM/GMM/AEC
Valve	Cooler	100 %/100 %/100 %/100 %	100 %/82 %/100 %/100 %	0 %/62 %/0 %/0 %
Pump	Cooler	100 %/100 %/100 %/100 %	87 %/77 %/87 %/87 %	35 %/72 %/35 %/37 %
Accumulator	Cooler	100 %/100 %/100 %/100 %	77 %/77 %/77 %/77 %	100 %/100 %/100 %/100 %
Cooler	Valve	100 %/100 %/100 %/100 %	100 %/98 %/100 %/100 %	100 %/100 %/100 %/100 %
Pump	Valve	98 %/100 %/100 %/99 %	71 %/66 %/70 %/99 %	95 %/95 %/95 %/11 %
Accumulator	Valve	98 %/100 %/100 %/97 %	67 %/58 %/86 %/71 %	77 %/100 %/34 %/71 %

detected as normal, i. e. the samples with additional variations that are still correctly classified by the supervised classifier and are identified as normal by the novelty detection. 100 % indicates the desired case where none of the samples with additional variations which do not affect the supervised classification are detected as novel. Third, the table shows the percentage of samples misclassified by the supervised classifier which are correctly identified as novelty, i. e. affecting the supervised classifier; again, 100 % would be the desired rate.

In Figure 5 (a) the target is the classification of accumulator pre-pressure loss (shown by color) where only data for cooler working at 100 % is available during training, i. e. cooler variations are treated as novelties that can affect the supervised learning. Again, the automated toolbox [2] is used for supervised learning. Introducing cooler variations the cross-validated classification rate of the resulting classifier drops from 100 % for the training data (cooler working at 100 % DC) to only 15 % when the cooler is working at 20 % or 3 % DC. The target for novelty detection is therefore to detect this decrease in classification performance. First of all, note that independent of the specific novelty detection algorithm used all training data samples are assigned to the normal class and, thus, the number of false positives is zero. At the same time all samples misclassified by the supervised classifier are correctly identified as novel. However, only 77 % of the samples that are still classified correctly by the supervised approach despite the cooler variation are identified as normal. Therefore, in this case the novelty detection provides an indication for the user that a novel influence such as the cooler variation reduces the classification rate, in this case from 100 % to 15 %.

In the second example shown in Figure 5 (b), the classification rate of the accumulator pre-pressure is reduced from 100 % to 25 % due to the degradation of the valve switching characteristics. As in the first case all novelty detection algorithms can identify the training data as normal with a maximum of 2.7 % identified as novel by AEC. However, the percentage of correctly identified novel samples under valve variation varies between 34 % (GMM) and 100 % (SVM). Therefore, SVM offers the best performance in this application to detect the superimposed fault. However, in a real application scenario the novel fault scenario as well as the percentage of misclassified samples that are correctly identified as novel are unknown and the user thus has no chance to select the best algorithm for novelty detection. Therefore it is suggested to use ensemble methods, i. e. to run multiple novelty detection algorithms simultaneously, and identify a novelty, if any of the algorithms assigns the new data sample to the novel group.

5 Detecting unknown faults

A third and highly relevant application for novelty detection is the detection of so far unknown new faults, even if they would not disturb supervised classification of known faults. This scenario is motivated by the availability of data samples from machines working at optimal performance (normal condition of a new machine) while representative data samples from different fault conditions are absent in

industrial application scenarios and especially in retrofit solutions for data based condition monitoring. The idea is to start with data samples of the new machine in normal condition and to apply novelty detection. Once a novelty is detected, e. g. a machine fault, it can be analyzed and annotated by the maintenance crew. This identified fault can then be added to the training dataset for supervised classification allowing the system to automatically recognize this fault if it occurs again. At the same time the novelty detection algorithm is retrained so that it does not recognize this fault as novelty anymore since it is now known. This approach allows to successively build a comprehensive ML model of a machine or process during its operation. Note that a second but distinctly separate step would be to transfer this model to similar machines or processes.

In this application scenario not only threshold tuning and algorithm selection are critical to detect novelties while maintaining a low false positive rate. Another critical aspect is the feature representation of high dimensional raw data that has to capture the characteristics of the sensor data in low dimensional space to emphasize novelties and prevent overfitting [20] and the curse of dimensionality [17]. The approach taken is the following:

1. Extract features using the automatic feature extraction algorithm described in [2] providing the lowest approximation error on the training data of each individual sensor.
2. Standardize the extracted features to compensate scaling effects, i. e. to prevent large scale features from dominating the novelty score due to their large variance.
3. Perform PCA on the standardized features.
4. Train novelty detection using different algorithms. If novel data are available use receiver operating characteristic (ROC) plots to select the best algorithm.
5. Set the novelty threshold using histogram plots and group-based validation, so that left out groups are just recognized as normal to allow for interpolation. If no group affiliations are known an SVM with threshold = 0 is usually a good choice.

As in section 4 multiple application cases are simulated on the hydraulic dataset defining one or more of the four simulated faults as novelty and ignoring it during training. Keep in mind that this not only includes training of the novelty detection algorithm as in section 4 but now also training of the feature extraction. To illustrate this the data distribution when ignoring valve degradation is shown in Figure 6. In this example the valve degradation is clearly visible despite the fact that in the first two principal components, i. e. in the 2D visualization, only 26 % of total variance are shown and that the variations introduced by valve switching degradation were not part of the PCA calculation. This indicates that the chosen approach for feature extraction is capable of showing faults, even if their symptoms, i. e. delayed valve switching, are well localized within a split second after the switching and do not affect the overall characteristics of the measurement.

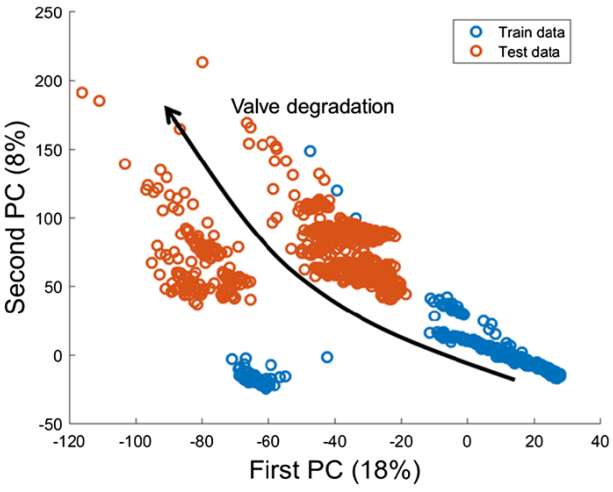


Figure 6: Principal Component Analysis plot used for training of the novelty detection algorithms. Features are extracted using only samples with the valve operating at 100 % using the automated approach described before. Although variations due to valve switching degradation are ignored during PCA calculation (training data) they are clearly visible in the test data. The variations seen in the training data originates from cooling power degradation.

Figure 7 shows the novelty scores achieved when training KNN on the data shown in Figure 6. For the training data scores validated with 10-fold cross validation are shown. As expected from Figure 6 KNN provides a high contrast between samples with the valve operating at 100 % (training data) and with valve switching degradation, i. e. valve operating at less than 100 % (test data). Since most training samples are concentrated in the dense cluster in the lower right corner of Figure 6 (the other clusters are cycles with valve switching failures, i. e. outliers, cf. section 3) Figure 7 (b) barely changes if group-based cross validation on one of the other faults is used instead of 10-fold cross validation. Therefore, even without access to the test data, a user would set the novelty threshold at 100 providing excellent recognition rates for the test data, i. e. degraded valve switching, while only identifying outliers as false positives.

To compare the different novelty detection algorithms, Figure 8 shows the histograms of similarity scores generated on the data from Figure 6 for GMM (a) and SVM (b).

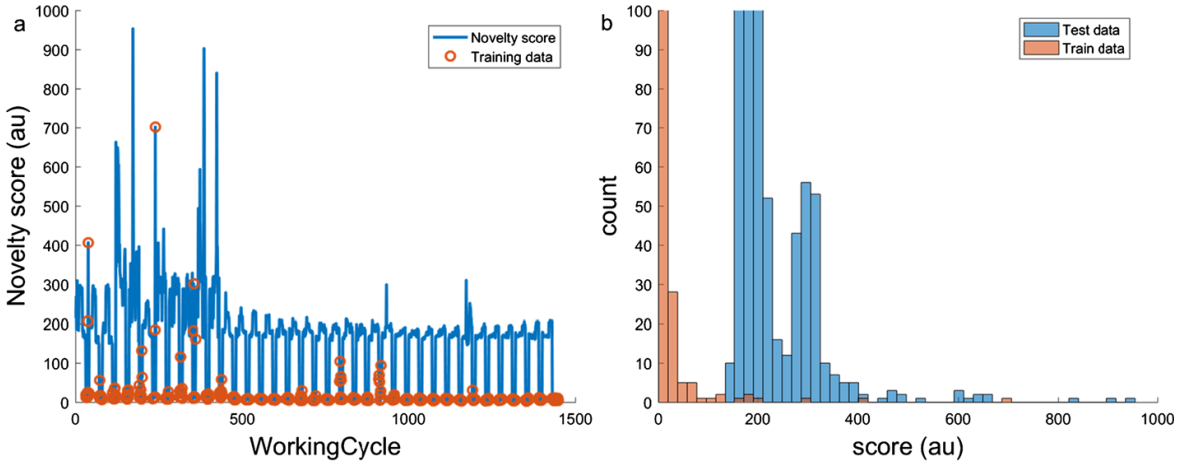


Figure 7: Progression plot (a) and histogram (b) of novelty scores computed by KNN novelty detection on the PCA shown in Figure 6. Outliers discussed in Section 3 and variations introduced by valve switching degradation (cf. Figure 2 (a); note that this is repeated for three different stages of cooling power degradation) are clearly visible. Scores on training data are reported as results of 10-fold cross validation and show clear separability between normal and novel data (outliers are accepted as novel).

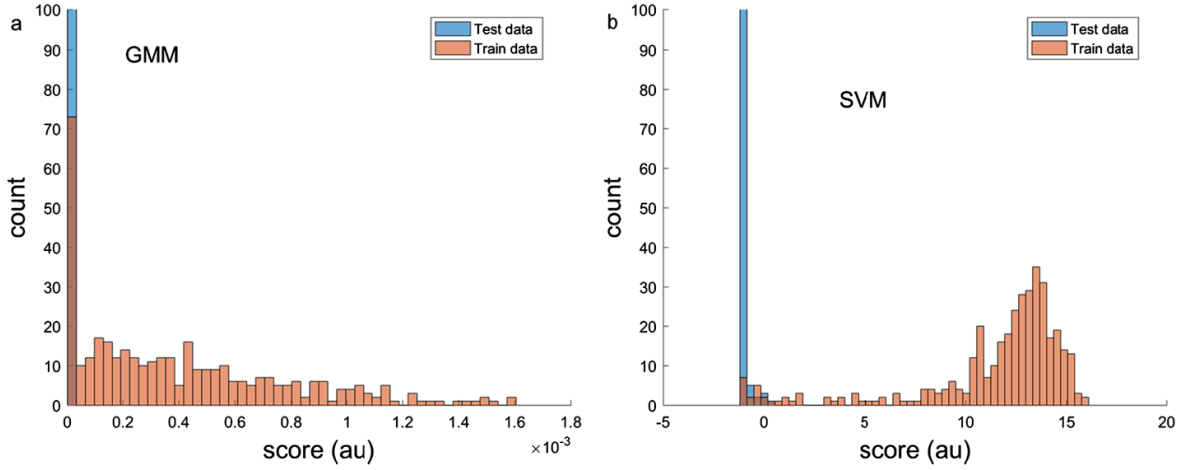


Figure 8: Histogram of novelty scores computed by GMM (a) and SVM (b) on the PCA shown in Figure 6. Outliers discussed in Section 3 are clearly visible for SVM in data samples with score < 2.5 that are distinctly separated from the bulk of training data samples with score > 2.5. Scores on training data are reported as results of 10-fold cross validation.

Note that both GMM and KNN show the outliers seen in Figure 6 outside the dense clusters or even overlapping with the test data. They are the same outliers seen in Figure 3. As discussed in Section 3, the exponential decrease of the GMM novelty score and the saturation of the SVM scores, respectively, make threshold tuning difficult compared to KNN. Furthermore, the KNN threshold is easier to interpret and should therefore be preferred in this scenario.

Similarly, KNN offers the highest contrast in almost all other scenarios, in which all possible combinations of fault variations had been removed from the training data. These results are in agreement with [21] where is was concluded that simple methods like KNN are more versatile than more sophisticated methods, which, however, can be better adjusted to individual application cases. For use by machine experts without extensive experience in ML methods as targeted in this paper simple methods without hyperparameter tuning are clearly favored.

Although KNN provides the best contrast between training and test data it is not capable of detecting all possible faults under all circumstances. Only novel faults which introduce significant new variations are detected reliably. Accordingly, cooler and valve degradation are always detectable as novelty since these two faults introduce the most significant changes in the sensor response, while accumulator faults are only detected at severe pre-pressure

loss and internal pump leakage is, in fact, not detected at all. However, this is a basic drawback of novelty detection since non-significant new variations are not expected to be assigned high novelty scores because this would result in a high false positive rate, which is of course not desirable. To detect these faults supervised learning methods that emphasize even small differences need to be employed and trained with suitable experimental simulations.

6 Discussion and outlook

This contribution discussed simple approaches to novelty detection that can be employed by engineers and machine experts for outlier detection, monitoring of supervised learning and detection of unknown faults without requiring in-depth knowledge of ML algorithms, mainly because little or no hyperparameter tuning is required. Note that the detection of unknown faults also includes the detection of sensor faults [22] to improve the robustness of the overall system. For intuitive application and user acceptance visualization is emphasized; all algorithms and plots are available as part of an open-source toolbox with graphical user interface [6]. It was shown that the described systematic approaches achieve reasonable novelty thresholds from training data alone. However, the user has to be aware that none of the approaches can solve the basic challenge, i. e. that novelty detection can only detect novelties which introduce variations in the data that significantly differ from the training data. This problem is aggravated further by the application of feature extraction and especially supervised feature selection which can prevent the detection of novel signal characteristics which were not present in the training data and which do not affect the characteristics captured by the features. On the other hand, in most application cases this is a necessary limitation due to the high dimensionality of the data which often prevents application of novelty detection directly on the raw data.

Future work will address the effect smaller steps in fault severity simulation have on the threshold chosen from group-based cross validation. Since this reduces the variations between severity levels that determine the chosen threshold smaller steps could allow a lower novelty threshold and therefore more sensitive novelty detection. In this paper, a variety of different novelty detection scenarios was evaluated, but all were based on the same dataset and experimental set-up. In the future, the described approaches will be tested on other datasets from different domains to verify the findings and to generalize the approaches.

Acknowledgment: The authors thank Pragya Pande for her overview of novelty detection [16] and for making her implementation of the used novelty detection algorithms available. Also the authors thank Jannis Morsch for implementing the reconstruction error based algorithm selection for feature extraction used in Section 5 and for porting the novelty detection algorithms and their visualizations to DAV^{3}E.

Funding: The research presented in this paper was in part performed during the projects MoSeS-Pro and iCM Hydraulics at ZeMA – Center for Mechatronics and Automation Technology gGmbH; MoSeS-Pro: German Federal Ministry of Education and Research, funding code 16ES0419K; iCM Hydraulics: EFI program (support of development, research, and innovation in Saarland), research by ZeMA was financed by HYDAC Filter Systems.

References

1. "Industrial Internet Consortium." [Online]. Available: http://www.iiconsortium.org. [Accessed: 10-Jun-2019].
2. T. Schneider, N. Helwig, and A. Schütze, "Industrial condition monitoring with smart sensors using automated feature extraction and selection," *Meas. Sci. Technol.*, vol. 29, no. 9, p. 094002, Sep. 2018.
3. P. Henriquez, J. B. Alonso, M. A. Ferrer, and C. M. Travieso, "Review of automatic fault diagnosis systems using audio and vibration signals," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 44, no. 5, pp. 642–652, 2014.
4. M. A. F. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Processing*, vol. 99, pp. 215–249, 2014.
5. E. Keogh and S. Kasetty, "On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration," *Data Min. Knowl. Discov.*, vol. 7, no. 4, pp. 349–371, 2003.
6. M. Bastuck, T. Baur, and A. Schütze, "DAV^{3}E – a MATLAB toolbox for multivariate sensor data evaluation," *J. Sensors Sens. Syst.*, vol. 7, no. 2, pp. 489–506, Sep. 2018.
7. M. Markou and S. Singh, "Novelty detection: a review – part 1: statistical approaches," vol. 83, pp. 2481–2497, 2003.
8. D. Miljković, "Review of Novelty Detection Methods," *Proc. Int. Conv. MIPRO*, no. June, pp. 593–598, 2010.
9. C. Chow, "On optimum recognition error and reject tradeoff," *IEEE Trans. Inf. Theory*, vol. 16, no. 1, pp. 41–46, Jan. 1970.
10. C. Keribin, "Consistent Estimate of the Order of Mixture Models," *Sankhy, Ser. A*, vol. 64, no. 01, pp. 49–66, Jul. 2000.
11. T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York, NY: Springer New York, 2009.
12. M. Zhao and V. Saligrama, "Anomaly Detection with Score functions based on Nearest Neighbor Graphs," *Neural Information Processing Systems*, pp. 1–9, 2009.
13. A. DasGupta, *Probability for Statistics and Machine Learning*. New York, NY: Springer New York, 2011.

14. A. Beghi, L. Cecchinato, C. Corazzol, M. Rampazzo, F. Simmini, and G. A. Susto, "A One-Class SVM Based Tool for Machine Learning Novelty Detection in HVAC Chiller Systems," *IFAC Proc. Vol.*, vol. 47, no. 3, pp. 1953–1958, 2014.
15. B. B. Thompson, R. J. Marks, J. J. Choi, M. A. El-Sharkawi, Ming-Yuh Huang, and C. Bunje, "Implicit learning in autoencoder novelty assessment," in *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)*, pp. 2878–2883.
16. Pragya Pande, "Novelty detection for unknown faults in industrial condition monitoring and further applications," Saarland University, 2018.
17. K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When Is 'Nearest Neighbor' Meaningful?," in *7th International Conference on Database Theory – ICDT'99*, 1999, pp. 217–235.
18. N. Helwig, E. Pignanelli, and A. Schutze, "Condition monitoring of a complex hydraulic system using multivariate statistics," in *2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, 2015, pp. 210–215.
19. D. Dua and C. Graff, "UCI Machine Learning Repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml. [Accessed: 10-Jun-2019].
20. K. Backhaus, B. Erichson, W. Plinke, and R. Weiber, *Multivariate Analysemethoden: Eine anwendungsorientierte Einführung, 9 überarbe*, Berlin Heidelberg New York: Springer-Verlag, 2000.
21. X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh, "Experimental comparison of representation methods and distance measures for time series data," *Data Min. Knowl. Discov.*, vol. 26, no. 2, pp. 275–309, Mar. 2013.
22. N. Helwig, E. Pignanelli, and A. Schütze, "Detecting and Compensating Sensor Faults in a Hydraulic Condition Monitoring System," in *Proceedings SENSOR*, 2015, pp. 641–646.

Bionotes

Tizian Schneider
Zentrum für Mechatronik und Automatisierungstechnologie ZeMA, Aktorik und Sensorik, Eschberger Weg 46, Gewerbepark, Gebäude 9, 66121 Saarbrücken, Saarland, Germany
t.schneider@zema.de

Tizian Schneider studied Microtechnologies and Nanostructures at Saarland University and received his Master of Science degree in January 2016. Since that time he has been working at Centre for Mechatronics and Automation Technology (ZeMA), division 'Sensors and Actuators' and leads the field of data-based condition monitoring for industrial applications such as fluid power and electromechanical drive systems.

Steffen Klein
Universität des Saarlandes, Lehrstuhl für Messtechnik, 66123 Saarbrücken, Saarland, Germany
s.klein@lmt.uni-saarland.de

Steffen Klein studied Mechatronics and received Master of Science degree in May 2018. Since that time he has been working at Saarland University at the Laboratory for Measurement Technology (LMT), in the field of data-based condition monitoring for industrial applications such as fluid power and electromechanical drive systems.

Andreas Schütze
Universität des Saarlandes, Lehrstuhl für Messtechnik, 66123 Saarbrücken, Saarland, Germany
schuetze@lmt.uni-saarland.de

Andreas Schütze received his diploma in physics from RWTH Aachen in 1990 and his doctorate in Applied Physics from Justus-Liebig-Universität in Gießen in 1994 with a thesis on microsensors and sensor systems for the detection of reducing and oxidizing gases. From 1994 until 1998 he worked for VDI/VDE-IT, Teltow, Germany, mainly in the fields of microsystems technology. From 1998 until 2000 he was professor for Sensors and Microsystem Technology at the University of Applied Sciences in Krefeld, Germany. Since April 2000 he is professor for Measurement Technology in the Department of Mechatronics at Saarland University, Saarbrücken, Germany and head of the Laboratory for Measurement Technology (LMT). His research interests include microsensors and microsystems, especially intelligent gas sensor systems for security applications.

6 Subsequent Research Lead by Author

The successful research and demonstration of the automated machine learning toolbox in the research project MoSeS-Pro formed the foundation of the joint Data Engineering and Smart Sensors (DESS) research group at ZeMA and Saarland University. This group was funded by multiple further research projects and project participation based on the toolbox developed in MoSeS-Pro (see Table 1). It is led by the author of this thesis. This continued work is an important part of this thesis and shows the great research interest in this field.

As of December 2023, it comprises nine (temporarily ten) full-time scientists. The following scientists at least partially base their doctoral dissertations and scientific work on the automated machine learning toolbox described in Papers 1-3:

- **Dr. Tanja Dorst** researched how uncertainty in automated machine learning for condition monitoring can be expressed following the GUM (see papers C and D). She developed a metrological framework providing analytical estimates of measurement uncertainty for all algorithms in the automated machine learning toolbox [120].
- **Christopher Schnur** researched whether and how the principles and algorithms suggested in this dissertation can be transferred to structural health monitoring (see Paper E). Furthermore, he is working on how better data can be generated for automated machine learning in condition monitoring and has created a guideline for projects that utilize automated machine learning [121].
- **Steffen Klein** focuses on how to apply automated machine learning and especially novelty detection in retrofitting scenarios for machines that are running in continuous production and do not allow any experimental control for training data acquisition.
- **Payman Goodarzi** focuses on potential solutions for problems caused by domain shifts. He extensively benchmarked the toolbox's algorithms [122] and showed their favorability in group-based validation scenarios [56].
- **Yannick Robin** focused on how to apply automated machine learning to calibration problems of semiconductor gas sensors and compared it to neural networks [123, 124].

- **Christian Fuchs** researches how labeling errors affect the performance of automated machine learning [125] and how to choose a suitable hardware platform for smart sensors.
- **Sebastian Pültz** researched how to combine different types of feature extraction in a single model [85]. He also studied how to specialize this thesis's principles to monitor the conditions of helical gears [125].
- **Julian Schauer** works on how generic neural network accelerators can infer non-neural network models. He created neural network representations of the automated machine learning toolbox hardware-accelerated and energy-efficient inference [114].
- **Houssam El Moutaouakil** researches how machine learning can be used to localize defects in composite materials of hydrogen pressure tanks with ultrasonic guided waves independent of interfering effects [126].

6.1 Paper B: Comparison of Different ML Methods Concerning Prediction Quality, Domain Adaptation and Robustness

Further evaluations on multiple datasets revealed that domain shifts pose a primary challenge for ML-based condition monitoring and can significantly reduce damage detection and quantification performance. Those issues can only be identified by directly testing for robustness against change in the causative variable by leave one out cross-validation or holdout method. The causative variable can be anything that influences sensor data, like different operation modes, different individual machines, variations in processed materials, or ambient conditions. The extent of the imposed domain shift and its impact on classification and regression might vary depending on the variable. It might range from neglectable to model-breaking. Also, the notion of domain shift is not limited to constant offsets but might also result in linear or non-linear distortion of the patterns learned from training data. Domain shifts can be both seen as a robustness problem and a transferability problem.

Since the domain shift problem has to be solved for the widespread application of ML-based condition monitoring, the following paper explores the robustness of different algorithms against domain shifts and simple methods for domain adaptation, like offset calibration and compensation. The results are shown for classification and regression. They include the previously described automated machine learning toolbox, an additional feature extraction method, SVM, and different neural network architectures. In the shown examples, even simple compensation methods can compensate large parts of the deteriorated performance under domain shifts. Additionally, it can be seen that classical machine learning approaches based on feature extraction, selection, and non-neural network classification/regression outperform neural networks before and after domain adaptation in the given examples.

Comparison of Different ML Methods Concerning Prediction Quality, Domain Adaptation, and Robustness

Payman Goodarzi[1], Andreas Schütze[1], and Schneider Tizian[1]

[1]*Saarland University, Lab for Measurement Technology, Saarbrücken, Germany*

tm – Technisches Messen (2022), 89 (4), 224-239

The original paper can be found online at https://doi.org/10.1515/teme-2021-0129.

© Used with permission of Walter de Gruyter and Company, from Comparison of Different ML Methods Concerning Prediction Quality, Domain Adaptation and Robustness, Goodarzi, Payman; Schütze, Andreas; Schneider, Tizian, 89, 14, 2022; permission conveyed through Copyright Clearance Center, Inc.

Payman Goodarzi*, Andreas Schütze, and Tizian Schneider

Comparison of different ML methods concerning prediction quality, domain adaptation and robustness

Vergleich verschiedener ML-Methoden bezüglich Vorhersagequalität, Domänenanpassung und Robustheit

https://doi.org/10.1515/teme-2021-0129
Received December 14, 2021; accepted February 3, 2022

Abstract: Nowadays machine learning methods and data-driven models have been used widely in different fields including computer vision, biomedicine, and condition monitoring. However, these models show performance degradation when meeting real-life situations. Domain or dataset shift or out-of-distribution (OOD) prediction is mentioned as the reason for this problem. Especially in industrial condition monitoring, it is not clear when we should be concerned about domain shift and which methods are more robust against this problem. In this paper prediction results are compared for a conventional machine learning workflow based on feature extraction, selection, and classification/regression (FESC/R) and deep neural networks on two publicly available industrial datasets. We show that it is possible to visualize the possible shift in domain using feature extraction and principal component analysis. Also, experimental competition shows that the cross-domain validated results of FESC/R are comparable to the reported state-of-the-art methods. Finally, we show that the results for simple randomly selected validation sets do not correctly represent the model performance in real-world applications.

Keywords: Machine learning, condition monitoring, domain adaptation, neural network.

Zusammenfassung: Machine Learning und datenbasierte Modelle sind in der Literatur zu Computer Vision, Biomedizin oder Zustandsüberwachung weit verbreitet. Allerdings zeigen diese Methoden oft Schwächen in der realen Anwendung. Domain Shift oder Vorhersagen außerhalb der Verteilung der Trainingsdaten werden häufig als Ursache benannt. Besonders bei industrieller Zustandsüberwachung ist unklar, wann diese Probleme auftreten und welche Algorithmen robust dagegen sind. In diesem Beitrag werden die Ergebnisse einer klassischen ML-Auswertekette bestehend aus Merkmalsextraktion, Merkmalsselektion und Klassifikation bzw. Regression (FESC/R) mit jenen von mehrschichtigen neuronalen Netzen auf zwei öffentlich verfügbaren Datensätzen verglichen. Es wird gezeigt, dass mögliche Datenverschiebungen mittels Merkmalsextraktion und Hauptkomponentenanalyse sichtbar gemacht werden können. Weiterhin wird gezeigt, dass die mit FESC/R auf Domain Shift Problemen erreichten Ergebnisse gleichwertig zu denen von mehrschichtigen neuronalen Netzen sind. Letztlich wird gezeigt, dass eine zufällige Kreuzvalidierung die in einer realen Anwendung zu erwartende Genauigkeit eines ML-Modells nicht hinreichend abbilden kann.

Schlagwörter: maschinelles Lernen, Zustandsüberwachung, Domänenadaption, neuronale Netze.

1 Introduction

Condition monitoring and predictive maintenance are important applications for machine learning (ML) algorithms. Input data in these applications comes from different industrial sensors, e. g., pressure, temperature, vibration, or microphones. Targets for these tasks are usually predicting fault types, remaining useful lifetime (RUL), or detecting anomalies. Detecting faults or anticipating upcoming failures can significantly reduce downtime of industrial systems and furthermore ensure the quality of products [1], therefore more and more companies start to invest in predictive maintenance systems that are more applicable within the framework of Industry 4.0 [2].

The performance of modern data-driven models depends on the quality and quantity of supplied observations, however achieving proper data that covers all possible variations of a system and its environment to train

***Corresponding author: Payman Goodarzi,** Universität des Saarlandes, Lab for Measurement Technology, 66123 Saarbrücken, Germany, e-mail: p.goodarzi@lmt.uni-saarland.de
Andreas Schütze, Tizian Schneider, Universität des Saarlandes, Lab for Measurement Technology, 66123 Saarbrücken, Germany, ORCID: https://orcid.org/0000-0003-3060-5177 (A. Schütze)

these models is costly. A proper design of experiment (DoE) should include different control conditions and multiple recordings of a single target in different process situations and environments, e. g., for a ball bearing and an attached vibration sensor all possible combinations of temperatures, load and speed levels, lubrication conditions, vibrations transmitted by other machinery and peculiarity of production tolerances. This is exacerbated further when taking outdoor applications into account, e. g., for hydraulic machinery, because of the wider temperature range and additional environmental factors. Usually, variables considered less important for a process or expensive to change are ignored or varied in a limited range or step size to limit experiment costs. Either control variables are discrete or continuous, a design of experiment can cover just a limited number of them and respectively subsets of the complete target space are available for training [3]. However, generalizing a model among these subsets is difficult because the control conditions and the environment can change the distribution of data and may result in an OOD problem and domain shifts [4].

Many real-life applications of ML for condition monitoring impose domain shift problems onto the algorithms and thereby decrease its performance. Supervised ML methods mostly rely on the assumption that both training and test data come from the same distribution. This distribution of data can be called a domain and ideally, there is only one domain in a supervised learning task [5]. As mentioned, in industrial applications it is highly likely that working conditions affect the data distribution. For instance, operating temperature, oil or air pressure, rotating speed are common operating conditions that can cause a significant shift in the data distribution. Consequently, usually in real-world scenarios we encounter OOD problems, where the source domain is different from the target domain.

In classical measurement science, changes in the environment (computer science: domain shifts) are tackled with calibration and adjustment of the measurement system which is also possible for machine learning algorithms. To perform adjustment of ML algorithms different algorithms and approaches are proposed. The work of Moreno et al. [5] is one of the first attempts to unify the concepts and nomenclature in this field, because before that many works had been published about the same concept but with inconsistent naming [6–10]. Former studies include multi-task learning [11], instance weighting [12], visual domain transformation [13], maximum mean discrepancy [12] while over the past decade most of the works have been based on deep learning models [14–19]. In the field of industrial datasets, some of the recent works in this field are conditional maximum mean discrepancy-based ANN [20], the virtual adversarial training and batch nuclear-norm maximization [21], adaptive batch normalization for networks with wide first-layer kernels [22], and multi-kernel maximum mean discrepancies in multiple layers [17]. A common element among the recent methods that have been proposed for the diagnosis of industrial applications under domain shift is using deep learning architectures while the comparison with conventional ML is missing. In this study we compare an approach using feature extraction, selection, and classification/regression (FESC/R) with ANNs in terms of robustness against the cross-domain shift using two publicly available datasets and study the effect of calibration and adjustment as a domain adaptation technique on the defined tasks.

The rest of the paper is structured as follows: Section 2 first introduces a dataset from a hydraulic machine representing a regression problem and a dataset on damage detection in a ball bearing representing a classification task. Both datasets comprise domain shifts that are visualized. Furthermore, Section 2 introduces the two ML approaches compared in this study, i. e., a more classical approach based on feature extraction, feature selection and classification/regression and a more modern approach based on neural network architecture search. Section 3 shows how classification and regression results are affected by domain shifts in the mentioned datasets and how calibration and adjustment can help to compensate those effects before the study is concluded in Section 4.

2 Material and methods

In this section, we introduce datasets and methods that are used in this study. Methods consist of ANNs and FESC/R which is based on conventional ML approaches. The two publicly available datasets are (1) a hydraulic system (HS) dataset from Center for Mechatronics and Automation Technology (ZeMA gGmbH) and (2) a bearing dataset from Case Western Reserve University (CWRU).

2.1 Datasets

2.1.1 ZeMA hydraulic system dataset

The first dataset used in this study is the recorded behavior of an HS where multiple common faults of such a system are simulated in a testbed [23]. This is a publicly available dataset [24] and includes the recording of 17 sensors over

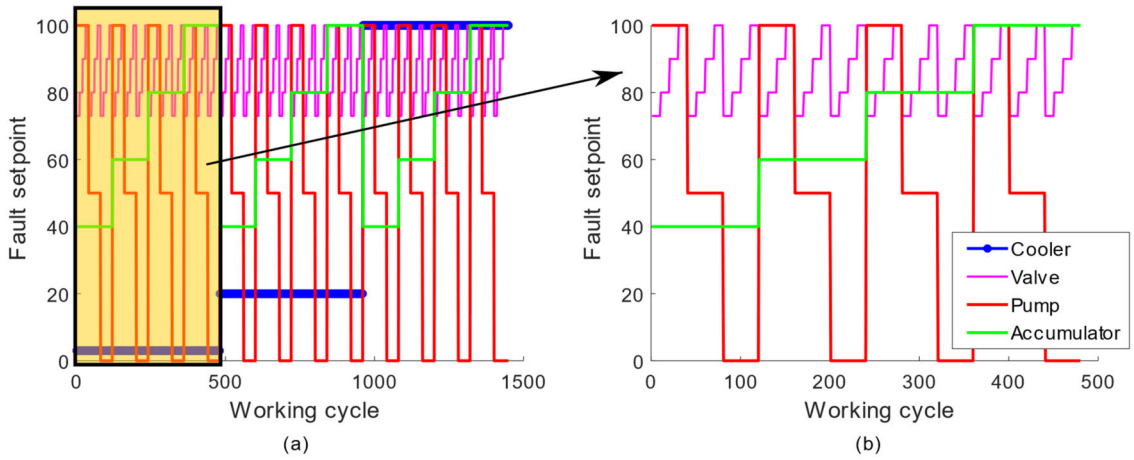


Figure 1: DoE of ZeMA dataset. All possible combinations of faults were repeated three times (a) for different cooler states, 100 % (normal operation), 20 %, and 3 % performance (set by varying the duty cycle of the cooler ventilator). The combination of faults for valve, pump, and accumulator is plotted in figure (b).

typical operating cycles with 60 seconds duration. Sensors measure process values like pressures, temperatures, and volume flows. Sampling frequencies for the sensors range from 1 to 100 Hz resulting in observations with 60 to 6000 data samples per sensor and cycle. The faults simulated in the ZeMA dataset are decreasing cooler performance, the main valve switching performance, internal pump leakage, and accumulator pre-charge pressure reduction. All fault conditions could be independently set by the control system. Figure 1 (a) shows conditions of the cooler, valve, pump, and accumulator in the dataset; Figure 1 (b) shows the systematic variation for valve, pump, and accumulator in more detail with these cycles being repeated three times for three cooler performances, 100 %, 20 %, and 3 %, respectively, which could also represent different climatic conditions or domains. We choose this dataset because changing the process conditions may lead to dataset shifts which is the main topic in this study. The training and validation scenarios are designed in a way that the final model should be robust against the cooler performance. We divided the data into training and test groups according to the cooler states. The training data includes cooler states of 100 and 20 percent, and test data is when the cooler worked just at 3 percent performance. The valve condition is considered as the target for the regression task.

The control variable with the biggest influence on the sensor data is the performance of the cooler. To show the influence of the process conditions on the distribution of data, we extracted statistical features of raw data using StatMom which is described in Section 2.2.1. Then, Principal Component Analysis (PCA) was applied on the extracted features, the results for the first two components are shown in Figure 2. As is evident from Figure 2, the cooler has a major influence on the data distribution and a change of the cooler performance results in a shift along the first principal component (PC), indicating the main source of variance in the dataset. Consequently, for this task the observations that belong to each cooler state can be considered as separate domains. Additionally, the cooler state is the most expensive control variable to change because after each change the machine has to run for several hours before a new temperature equilibrium is reached, and conditions are stable again [23]. Because the cooler state influences the machine's temperature and thereby the oil's viscosity its impact is evident in all measured sensor signals.

The learning scenario chosen for this dataset is the assessment of the current valve switching characteristic from 72 % (barely working) to 100 % under the condition that only data from cooler state 20 % (equivalent to 55 °C average temperature) and 100 % (equivalent to 44 °C average temperature) are used for training. Correctly predicting the valve characteristic at cooler state 3 % (equivalent to 66 °C average temperature) [23] would prove the model to be robust against environmental changes of temperature and is therefore the chosen ML task for the evaluated algorithms.

For calibration and adjustment of the models, data recorded at 3 % cooler state (new domain) and 100 % correct valve operation was considered. This is equivalent to using few measurements from a new machine (valve at 100 %) in a different environment for calibration and adjustment. The model is then evaluated on all data at cooler state 3 %.

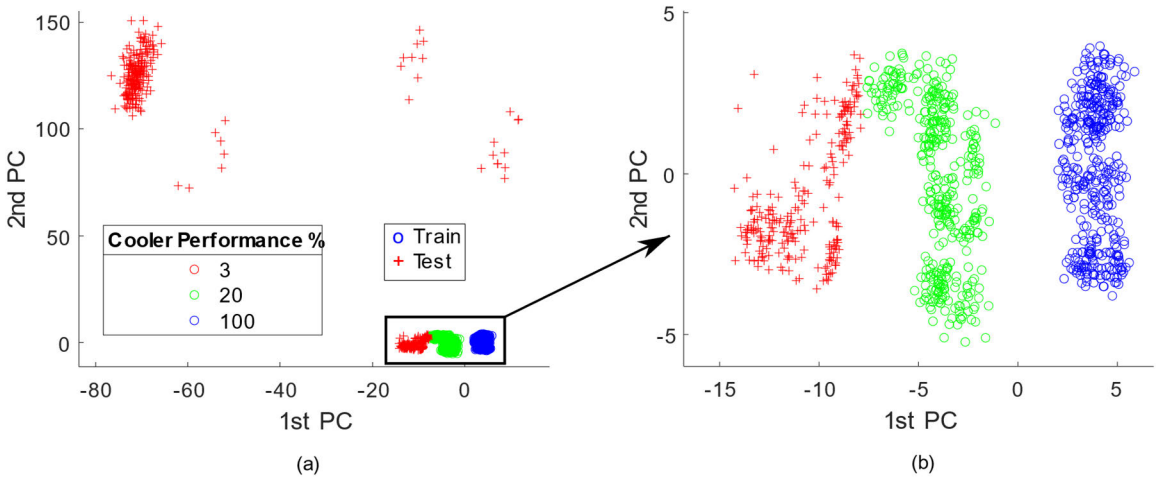


Figure 2: Features from the ZeMA dataset after PCA. (a) all data samples are colored by the cooler performance. (b) subset of observations that have more similarities. Shifts in the distribution due to the cooler changes are visible.

2.1.2 CWRU bearing dataset

The second dataset that is used in this study was published by the Bearing Data Center of Case Western Reserve University (CWRU) [25]. CWRU is a publicly available dataset that is frequently used in condition monitoring publications [22, 26]. Recorded data are vibration signals from the fan and drive ends of the testbed, the data is available in both 12 kHz and 48 kHz sampling rates. Four health states of the system are recorded, three different fault types and a healthy state without defects, the faults being damages at the inner and outer ring of the bearings and at the balls. Additionally, the process conditions were changed during the experiments; these process conditions are the rotational speed, fault diameters (0, 7, 14, and 21 mil corresponding to 0, 180, 360, and 540 µm) and motor load (0, 1, 2, and 3 hp corresponding to 0, 0.75, 1.5 and 2.25 kW). Note that we keep the original imperial units in the following instead of converting to SI units to avoid confusion when comparing our results with other evaluations for this widely used data set. A summary of the dataset is presented in Table 1. Although CWRU is extremely popular in the condition monitoring community, most of these studies have been done in different scenarios of the selected target and validation approach [26]. Predicting all combinations of fault types and fault sizes (10 classes) is a common scenario among the published studies, however the number of observations for each class is limited. We designed the scenario to cover the generalizability of the models on different load conditions by choosing the fault types as the classification target. In this study, the 48 kHz sampling version of recordings is used, and original recordings are cut into equal chunks with a length of 24k to have a constant number of data points in each observation.

Table 1: Summary of CWRU dataset.

Fault types	Fault size (mil)	Load (hp)	Rotational Speed (rpm)	Sensor Orientation
No Damage	0	0, 1, 2, 3	1725–1796	12
Inner Ring	7, 14, 21	0, 1, 2, 3	1721–1796	12
Outer Ring	7, 14, 21	0, 1, 2, 3	1723–1796	3, 6, 12
Ball	7, 14, 21	0, 1, 2, 3	1721–1796	12

To demonstrate domain shifts and domain adaptation in classification tasks, the learning scenario was chosen to be the detection of fault type (vs. fault severity). The four groups to be detected are damage at the outer ring (OR), inner ring (IR), ball (B) and no damage (None). In a real world application this detection should be possible independent of the load. Therefore, the training data was chosen to be the data recorded at 1, 2, and 3 hp load. The test data is the data recorded at 0 hp load respectively.

As in the ZeMA dataset we extracted features from the dataset, the result of a PCA performed on the extracted features is presented in Figure 3. In contrast to the ZeMA dataset, it is expected that the most relevant features come from the frequency domain of the vibration sensor. Therefore, a Time Frequency Extractor (TFEx, Section 2.2.1) was used for this use-case. Figure 3a shows the PCA plot colored to indicate different loads of the motor and Figure 3b visualizes the same data by coloring according to the damage target for the defined scenario. The healthy state, highlighted with an ellipse in both figures, shows a shift of the

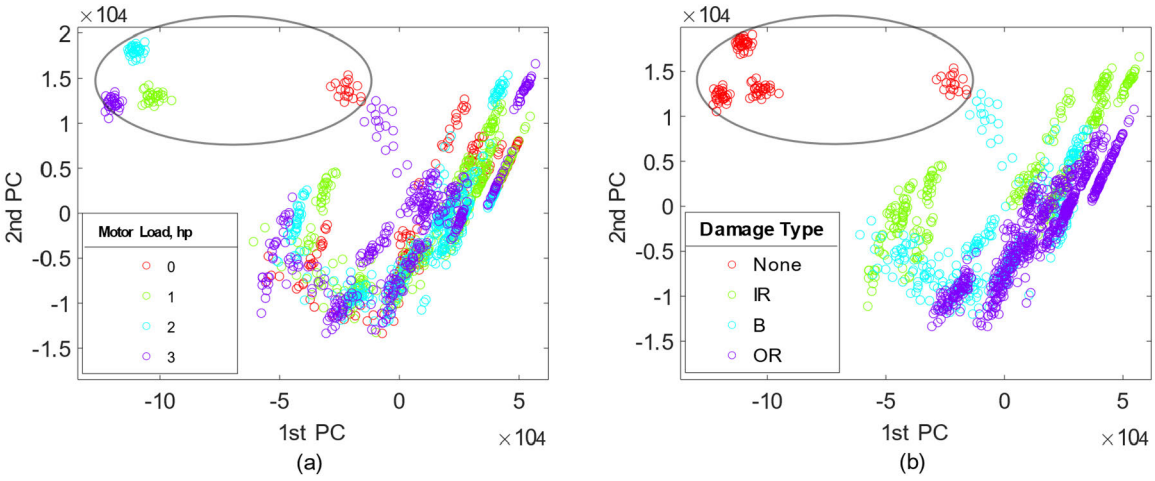


Figure 3: Features from the CWRU dataset after PCA. Visualizing the features based on the motor loads (a). Visualizing the features based on the damage types (b).

data for the motor at zero hp, which can cause difficulties for a model trained only on the other load conditions (1, 2, 3 hp). As this is the most obvious influence of process conditions on the data, it was chosen to be the test set in this scenario.

2.2 Algorithms

Various data-driven models have been applied in condition monitoring and predictive maintenance, including linear discriminant analysis (LDA) [27], support vector machines (SVM) [28], artificial neural networks (ANN) [29]. To make a comparison between methods, in this study two types of ML models are used. The first one is representing a more classical ML approach based on multivariate statistics using an automated approach to benchmark and choose the best algorithms (see Section 2.2.1) and the second one is targeting the increasingly popular deep learning models utilizing neural network architecture search for end-to-end learning (see Section 2.2.2).

2.2.1 FESC/R

Conventional ML methods have been used for a long time [30] and are still popular and effective in industrial applications [31]. One of the advantages of conventional ML models over neural networks is their explainability and interpretability. It means that decisions and predictions of a model can be explained in a human understandable manner, as a result the model would be more reliable and trustable [32, 33]. Therefore, conventional ML is preferred in fields where the model safety is critical and fault diagnosis is needed, like industrial and medical applications. On the other hand, these methods include explicit feature engineering and sometimes also have problems nowadays in handling big datasets.

We can formulate the conventional ML methods in form of a pipeline that consists of feature extraction (FE), feature selection, and classification (FESC) or regression (FESR). Depending on the model and input dimensions, it is also possible to apply a classifier/regression directly on the raw data, but in general FE methods are needed to reduce the dimensionality of the data. FE methods are usually necessary for condition monitoring applications because the raw data can be high-dimensional inputs [34], i. e., vibration signals from a bearing or current signals from a motor, and it would be difficult to find the relevant and key features directly from the raw data. These methods can extract general features independent of the use-case and observations or engineered features that are explicitly designed for a specific task. Deciding to use engineered features or using general FE methods depends on the resources, complexity of the task, and resource constraints of the use-case during training. For wide application, methods not requiring a trained data scientist are preferable, thus we will focus on general FE methods here which allow automation of the ML method adaptation [27]. Similarly, feature selection (FS) works for the same purpose to further reduce the data dimensionality, but usually makes use of supervised methods, while general FE is based on unsupervised methods. The goal is to reduce the number of features, decrease the complexity and improve

the performance of the model, especially to avoid overfitting.

In this study an open-source MATLAB toolbox [34] for conventional ML models was used, and FEFSC/R is mainly based on this publicly available toolbox. As described in [27] a fixed structure of methods is utilized to handle ML tasks. Although the overall structure is fixed, one of several different, mutually complementing algorithms is used for feature extraction and selection, respectively. This toolbox builds a stack from the predefined methods and searches through different combinations of methods as well as number of features to select the best-suited stack for the target task. As finding the best algorithms and hyperparameters (HP) in this framework is automatic, one of the main limitations of conventional machine learning, explicit feature engineering, is resolved. This framework shows a reliable performance in diverse applications ranging from industrial fault detection, remaining useful life (RUL) estimation, classification of human movement patterns to gas sensor systems [34–37], therefore this method was selected to study its behavior in inter-domain problems.

Here the focus is on showing these methods characteristics in OOD problems and the goal is measuring the robustness of the models in an OOD scenario. The toolbox is used to search for the best methods and HPs for both datasets then from the results the following methods are selected. The first FE function is called StatMom [27], which extracts the first four statistical moments (mean, variance, skewness, and kurtosis) from the input signal over defined time intervals. StatMom is simple and reliable, therefore it is our first candidate to reflect the general trend of a dataset. However, relying just on the time domain features is not sufficient in many use cases where the main information is contained in the frequency domain. Therefore, the second FE used in this study is the Time Frequency Extractor (TFEx) which extracts features from both time and frequency domains. TFEx extracts the root mean square (RMS), variance, linear slope, maximum, position of maximum, skewness, kurtosis, and peak to RMS ratio values from sections of both the time and frequency representation of input signals.

As FS we used two methods, namely Relieff [38] for the classification task and Pearson correlation for the regression task. After ranking features by the mentioned methods, a search for the best number of features to maximize the prediction accuracy is performed [27], then the selected features are transferred to the final block, i. e., classification or regression, of the training stack. Finally, the last element of the stack in this study applies a classification or regression method. The classification method in this study combines LDA and Mahalanobis distance (MD) to group mean [27] as classifier. LDA (also called Fisher linear discriminant analysis) [39] is a supervised ML approach that finds discriminant functions (DF) that maximize inter-class variance and minimize the intra-class variances. DFs of the LDA method are linear combinations of features and the best solution is guaranteed under the assumptions of identical class covariances and Gaussian distribution of the classes. MD is a simple but effective metric that measures the distance between a point and destination in a multi-dimensional space. Basically, MD is the Euclidean distance after transforming variables to remove the correlation and have unit variance, therefore it is not sensitive to the dimensions and units of data. For regression partial least squares regression (PLSR) [40] is used. The number of components for PLSR is chosen by an exhaustive search between one to the maximum number of features that feed to the PLSR method, the selection criterium is the best performance, i. e., lowest error.

Table 2: Features used in TFEx and StatMom.

TFEx (time and frequency domain)	StatMom (time domain)
RMS	Mean
Variance	Variance
Skewness	Skewness
Kurtosis	Kurtosis
Position of maximum	
Linear slope	
Maximum	
Peak to RMS ratio	

2.2.2 Deep learning methods

ANNs with three or more layers are called deep neural networks (DNN) therefore many modern network architectures are classified as deep learning methods. Over the past decade deep learning algorithms have been used in various applications and achieved outstanding results [41–43]. Researchers and developers utilize these methods in almost every purpose, i. e., autonomous driving, medical diagnosis, recommendation systems, translation, and predictive maintenance [26]. Conventional ML algorithms either have limited capacity or, when applied to big datasets, face difficulties during the training process. In contrast, ANNs are scalable and can be trained efficiently on big datasets with the help of the simple backpropagation algorithm. On the other hand, ANNs usually are used as black-boxes and explaining their predictions is difficult.

Table 3: List of HPs for CNN, including the search ranges. An iterative HP optimization approach is used and the ranges for the initial and final ranges reported.

HP	Initial trial	Final trial
Initial learning rate (log scale)	10^{-4}–10^{-2}	0.002
Kernel size	2–10	3–5
Depth	3–10 (Conv blocks)	5–10
# of neurons, fully connected layer	1–1000	1–100
# of filters	Fixed, relative to the depth	Fixed, relative to the depth
1^{st} convolutional layer filter size	10–100	10–35
Batch Size	32	32

Table 4: List of HPs for WaveNet-based network, including the search ranges. An iterative HP optimization approach is used and the ranges for the initial and final ranges reported.

HP	Initial trial	Final trial
Initial learning rate (log scale)	10^{-4}–10^{-1}	10^{-3}–10^{-2}
Kernel size	2–10	3–6
Depth	3–10 (Conv blocks)	3–10 (WaveNet blocks)
# of filters	8–100	40–80
1^{st} conv layer filter size	20–100	20–50

Designing and training DNNs requires tuning many hyper-parameters (HP). Hyper-parameters in ANNs can be categorized into two groups, the first one contains architecture HPs and the second learning HPs. Architecture HPs are parameters that specify the structure of a network, i. e., number of layers, filter size, number of filters in a layer, number of neurons in a layer, and of course the type of a layer. Training HPs specify the training process for a network when the architecture is fixed. Initial learning rate, mini-batch size, and number of epochs are examples for the training HPs. The process of choosing the best HPs is generally called HP optimization and more specifically for architecture HPs is named Neural Architecture Search (NAS) [44].

Although NAS showed particularly superior results outperforming human designed networks [45], most researchers in condition monitoring and predictive maintenance are using modified versions of published DNNs [29]. A systematic approach to report an experiment is to describe not only the methods, model, and architecture but also the HP optimization process that is used for the task. Otherwise, an unforeseen overfitting might be neglected because of non-linearity of DNNs, over-parametrization and the assumption that training, and test data come from the same distributions. In this study Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), Resnet [46] and WaveNet-style [47] networks were used in the NAS process for both defined tasks (ZeMA and CWRU datasets), however, we only report the architectures that achieved good validation accuracy for each scenario. In case of comparable error rates, the method with lower complexity is selected. NAS is a remarkably interesting field of study that is actively growing [45] and explaining its algorithms and methods are out of context of this study.

We used evolutional parametric architectures together with Bayesian optimization [48] as the NAS process [37]. This algorithm iteratively searches for the best HPs and in each trial the ranges for the parameters are adjusted according to the best results of the last trial and possible constraints, i. e., maximum depth or maximum number of filters. In the end, the network corresponding to the lowest validation loss is selected. The parametric architectures and the ranges of the parameters are explained in Tables 3 (for CNN) and 4 (WaveNet); Table 5 lists the HPs which were kept fixed during the experiments. These HP ranges were selected based on a work that used NAS to find a high-performance ANN for a similar dataset [37] and the original work of the WaveNet [47].

Two types of CNNs are used in this study: conventional CNN with a single forward path and a WaveNet-style [47] DNN using dilated convolutions [49] and skipped connections [46]. The simple CNN is used for the ZeMA dataset where the information is mostly contained in the time domain [23]. CNNs are the building block of many modern DNNs, moreover after multi-layer perceptron (MLP) networks CNNs are the simplest conventional network archi-

Table 5: List of fixed HPs during experiments.

HP	CNN	WaveNet-based
Batch Size	32	32
L2-regularization	0.001	0.001
Learn rate drop rate	0.9	0.9
Maximum epochs	100	10
Optimizer	ADAM	ADAM

tecture. Because of the extremely high capacity of MLPs, they are very prone to overfitting [50]. Therefore, MLPs are not chosen in this study and CNNs are the next simplest network architecture that is selected. While it is possible to also apply CNNs on complex high-frequency signals, the WaveNet-style network is used for the CWRU dataset, where the input is raw vibration signals, to add diversity to our study. Two losses according to the use cases are used, the mean squared error (MSE) for the regression task and the cross-entropy loss for the classification scenario.

2.2.3 Domain adaptation

As mentioned before, many ML approaches suffer from a degradation of the performance in real world scenarios due to a shift between training and test data [51]. Many algorithms have been developed to remedy this problem which can be categorized into different groups based on the training scenarios i. e., transfer learning, domain adaptation, and domain generalization. The idea is inspired by humans' approach to learning new tasks [52]. Although the idea of using transfer learning in ANNs training was first presented already in 1976 [53], it is actively used nowadays in deep learning applications. Transfer learning aims to build new knowledge based on a trained model which was trained in a different task or domain, e. g., applying the same form of feature extraction to train models for RUL detection of ball bearing in different sizes. Domain adaptation is a sub-category of transfer learning, where the task for source and target models are the same, but the domain is changing. An engineering example would be the detection of faults in a hydraulic system that is trained on a testbed and then transferred to an identical machine used in a different environment for which only few calibration data are available for the undamaged machine. Finally, the last member of this group is domain generalization, i. e., the ideal case where the trained model is not sensitive to the domains but relies only on features common to all domains. In the mentioned example of the hydraulic system this would require the model to be trained with data from many identical machines in different environments to identify features that are independent of the individual machine and environment. On the other hand, this model could be transferred to any additional machine without requiring further adaptation. Table 6 summarizes the characteristics of domain adaptation and similar methods.

Note that domain adaptation in ML is equivalent to the calibration and adjustment of conventional measurement systems. Both for ML methods and conventional measurements the deviation between the system output and a known target in few calibration measurements is used to adjust the output accordingly. This is typically done after a change in the environment (domain change) of the sensor system. Because both application examples shown in this paper can be interpreted as domain adaptation tasks the rest of this paper will focus on domain adaptation.

3 Experiments and results

In this section the results of evaluations for FESR/FESC and DNN models are reported side by side to allow easier comparison.

3.1 ZeMA hydraulic system, regression use-case

Although the target and other variables in this dataset are discrete numbers (due to restrictions concerning DoE), they represent continuous variables, and a model

Table 6: Domain adaptation compared with similar approaches.

	Source and target tasks	Source and target domains (joint distribution)	Access to target domain
Supervised Learning	Same	Same	–
Transfer Learning	Same/Different	Same/Different	Yes
Domain Adaptation	Same	Different	Yes, unlabeled, or limited labels
Domain Generalization	Same	Different	No

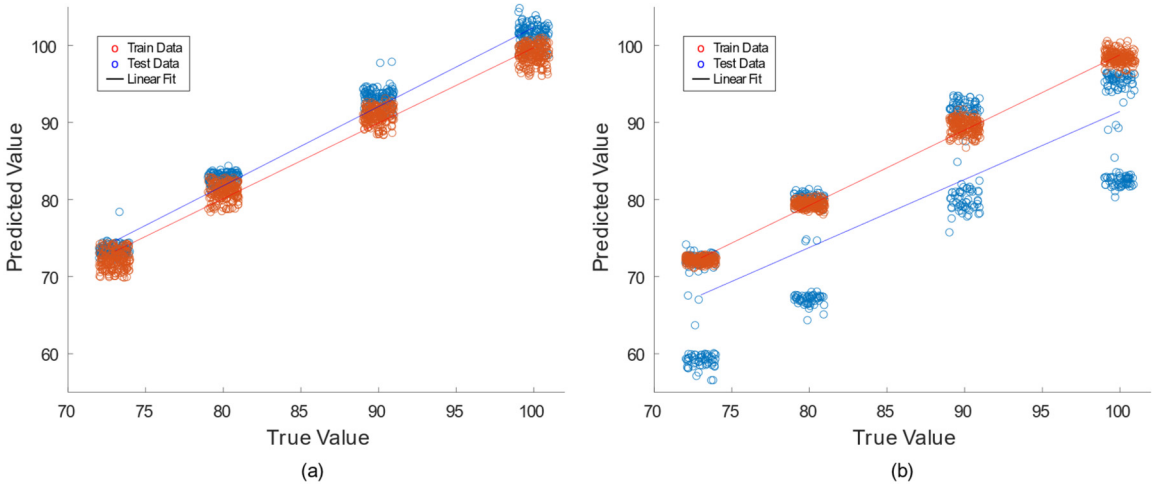


Figure 4: Prediction results for ZeMA dataset, linear lines show a fitted function on the training (red points) and test (blue points) predictions, to have a better visual representation a jitter plot is used. (a) Results from trained FESR stack. (b) Results from a trained CNN which is selected based on the validation loss.

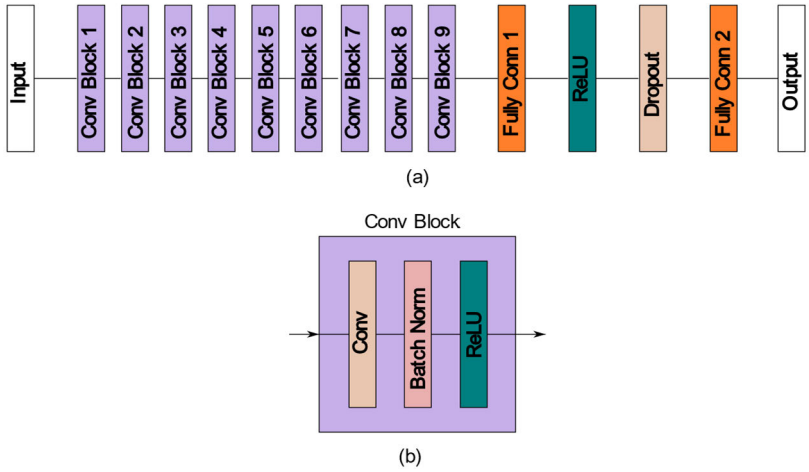


Figure 5: The output of the NAS algorithm for the ZeMA dataset (a). A convolutional block in this network consists of a convolution layer, a batch normalization layer and a ReLU layer (b).

should generalize over the complete ranges. The published dataset [24] has 17 different sensors, for simplicity we use the pressure sensor at the main valve (PS1 in the published dataset) that is corresponding to the defined target, the sensor has 100 Hz sampling rate.

3.1.1 The effect of domain shift on the trained models

In the earlier sections we illustrated the domain shift in the ZeMA dataset at the feature level. In this section we show the effect of this phenomenon when we train a model under this condition. The results are from two families of algorithms, FESR and deep learning models.

Starting with the FESR model, we trained a stack of selected methods for the defined task as described in Section 2.1.1. For the selected stack the FE method is StatMom, the FS is the Pearson correlation method, and finally PLSR is the last method of the stack. The results of predictions for the training and test data are plotted in Figure 4a. As there are just four discrete values in the targets a scatter plot with jittering[1] is used to provide a better view of overlapping data points – otherwise all samples would occur

1 Jittering is a simple but effective method to improve the visualization of discrete values in a scatter plot. By adding random noise to the observations, the overlapping points are separated. Note that it does not change the data permanently.

in four vertical lines and would be less distinguishable. Although the slope of the fitted linear lines for train and test data are similar, there is a clear shift between them. The change in temperature causes an offset error of approx. 2 %. This is equivalent to a conventional sensor system that suffers from a small cross-sensitivity to temperature. The root mean square error (RMSE) increases from 1.53 % (validation data) to 2.45 % (test data). The reason for this variation is the shifts of the distributions which was visualized in Figure 2; as the algorithms are not aware of the distribution of the test data, the shifts are not compensated. In the following we compare the results of a trained deep network for the same task.

Alternatively, we searched for a DNN architecture to fulfill the same task. The selected DNN is a 9-layer CNN as the outcome of the NAS algorithm with the architecture and parameters as reported in Figure 5 and Table 7, respectively, with the HPs ranges for the first and last trials of the search algorithm given in Table 3. The final ranges for the parameters are values that led to the best networks (with lowest validation losses) in earlier trials. Figure 6 shows the final trial of the NAS progress, each point in the plots is a trained model with the color representing the iteration number of the model from blue to yellow. Since the objective function of this process is the validation loss, the architecture corresponding to the lowest value was selected as the final model. However, the test RMSE of the resulting model is not as low as the validation RMSE, with validation and test errors of 1.15 % and 9.75 %, respectively. To explain why the trained network generalized so poorly on the test data, the predictions of the network for both training and test data are visualized in Figure 4b, also allowing direct comparison to the FESR model.

Table 7: Summary of parameters of the selected CNN after performing the NAS.

Layers	Filter Size (H × W)	Number of filters	Stride
Conv Block 1	1 × 20	8	1 × 3
Conv Block 2	1 × 4	8	1 × 2
Conv Block 3	1 × 4	16	1 × 2
Conv Block 4	1 × 4	24	1 × 2
Conv Block 5	1 × 4	32	1 × 2
Conv Block 6	1 × 4	40	1 × 2
Conv Block 7	1 × 4	48	1 × 2
Conv Block 8	1 × 4	56	1 × 2
Conv Block 9	1 × 4	64	1 × 2
Fully Conn 1	81	–	–
Dropout 50 %	–	–	–
Fully Conn 2	1	–	–

The deviation between the features of the source and target domains leads to a shift in the final predictions. While the selected network performs accurately on the validation data which are selected from the training distribution, it has difficulty in generalizing to the test data. As is evident in Figure 4b, the test data are divided into two groups, with one having a slight shift only from the training data but the second group being significantly shifted away leading to approx. 10 % error for the predictions. These two groups are visible also at the feature level in Figure 2a, where the test data consists of two separate groups. To allow a better visual representation of this problem, prediction results of the test data are plotted explicitly in Figure 7. The slope of the fitted line for both groups is almost identical but there is a clear offset between the two groups. Note that this problem would not be visible if a

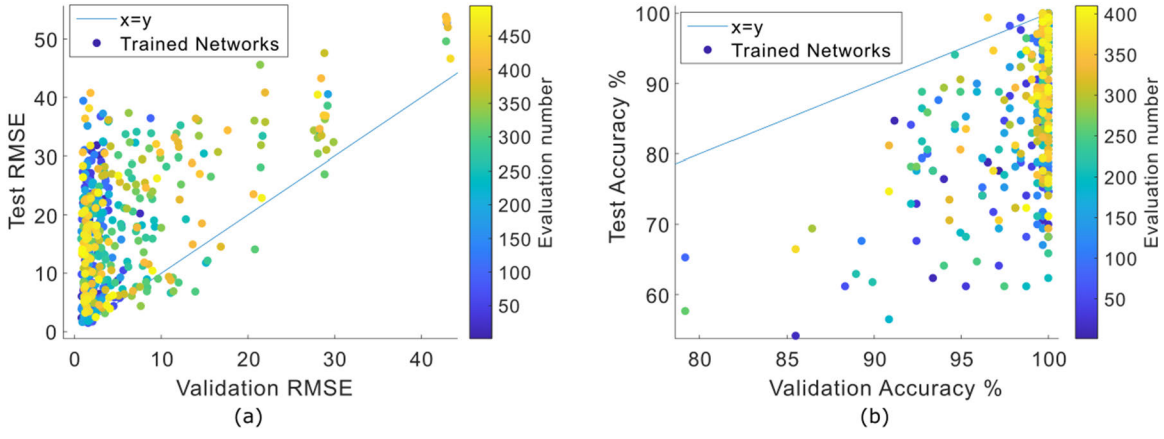


Figure 6: Final trial of the NAS algorithm. In this plot the validation data are 20 % of the training set which were randomly selected. The test data is from a different distribution, i. e., a different operating temperature. Each point is a trained network, (a) ZeMA use case, (b) CWRU use case.

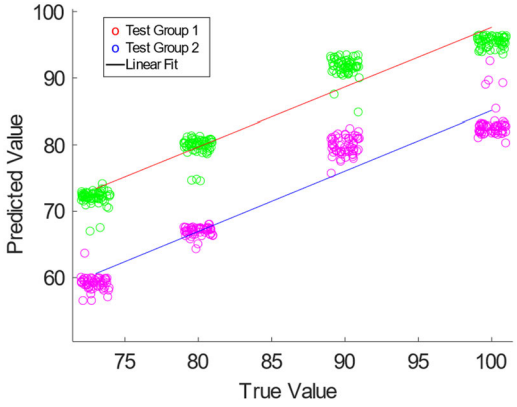


Figure 7: Predictions on the test dataset by the CNN. The "test group 1" are observations with a low error similar to the training data, while the "test group 2" are data with a significant shift regarding the target and thus high error.

simple random choice of test and training data had been used instead. Therefore, the validation scenario must be designed precisely to ensure covering cross-domain situations.

3.1.2 Domain adaptation

As shown in the last section, shifts in the dataset can significantly degrade the performance of a trained model on test data, especially if these represent a different domain. To reduce this problem and improve the results, calibration and adjustments are required. Calibration is performed using the test data of a single class (here: observations with 100 % performance) to simulate the real-world application of the previously trained model to a new machine that is working at 100 % but in an environment with a different temperature. As the simplest form of adjustment, the measured offset is removed in post-processing. Figure 8 shows the results after recalibration for both tested models, quantitative results are reported in Table 8. Recalibration for the ANN model is done just for the second test group (in Figure 7) that had a dominant shift with regard to the training data. While the results for both models improve with domain adaptation, FESR clearly yields a superior result with a test RMSE of 1.58 which is almost as low as the validation RMSE, while the RMSE of the CNN, although reduced by a factor 3, is still almost twice as high at 3.34.

3.2 CWRU, classification use-case

3.2.1 The effect of domain shift on the trained models

In the same way as for the HS use case, we first chose a stack of FESC that works best for this task. As mentioned above the FE method is TFEx (cf. Section 2.2.1), with Relieff used for FS and finally LDA and Mahalanobis distance for classification. The test accuracy for the test data is 99 %, which is exceptionally good. To check if the model compensated the shift for the test set, we visualize the projected features after the LDA. Figure 9a shows the results of

Table 8: Error rates for ZeMA dataset before and after recalibration.

Model	Validation RMSE	Test RMSE	Test RMSE after recalibration
FESR	1.53	2.45	1.58
CNN	1.15	9.74	3.34

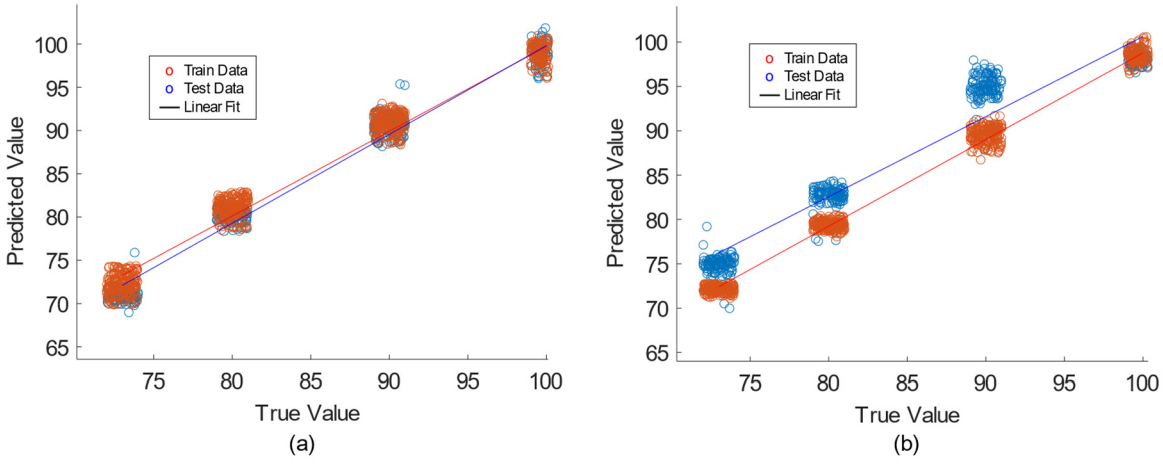


Figure 8: The FESR model predictions after recalibration (a). The CNN predictions after recalibration (b).

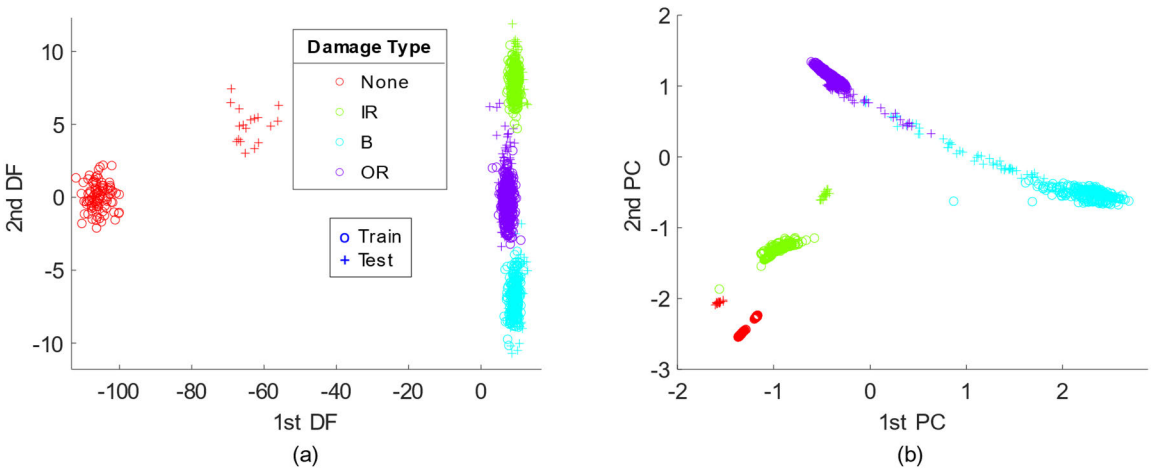


Figure 9: LDA projection of the features in the CWRU use case (a). PCA plot of the embedded features from the network in the last convolution layer, the graph shows the first and second principal components (PCs) of the features (b).

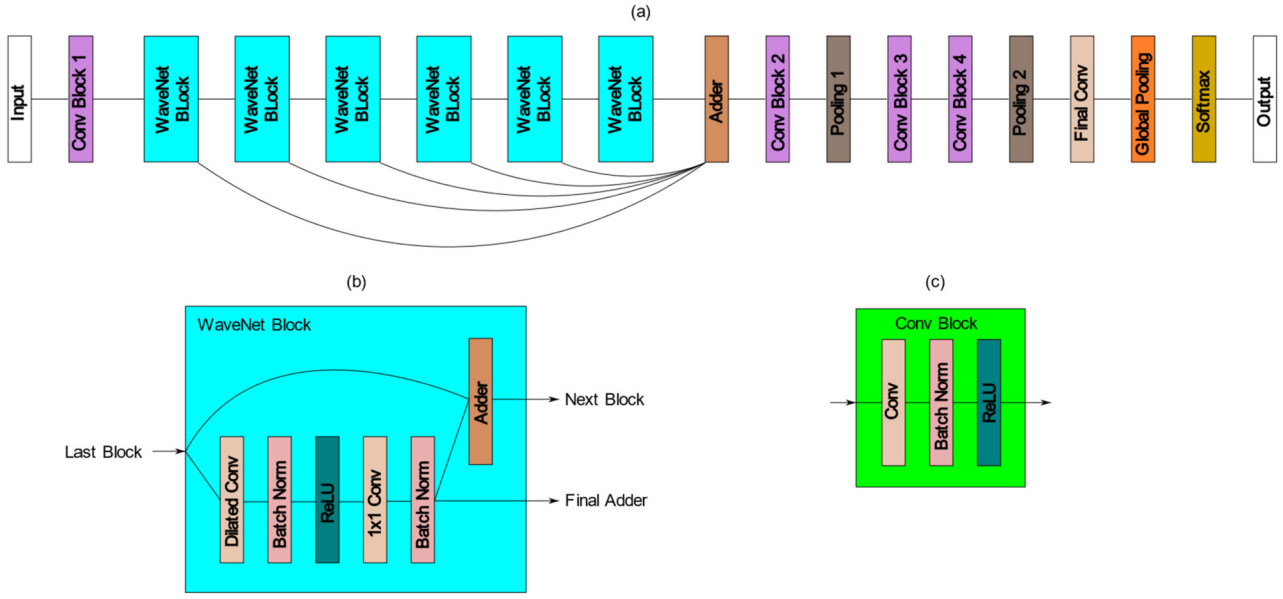


Figure 10: The WaveNet-based model with the lowest validation loss in the NAS algorithm (a). The WaveNet block (b) and a convolution block (c) that are used in the architecture.

the projection, which shows a small shift between training and test data for the damaged samples, but a significant shift for the healthy state (damage type "None"). However, the projections of those observations are still sufficiently far away from the other groups to be classified correctly. Also, it should be noted that the shifts are not in the same direction for all target groups, due at least in part to the fact that the targets are categorical and can therefore not be sorted in a logical order.

As mentioned above we expect relevant features also from the frequency domain for this use case, therefore a network architecture that previously showed superior results for raw audio and vibration signals, WaveNet, is used. An HPs search for the WaveNet-based network in accordance with Table 4 was conducted and resulted in the network shown in Figure 10 with HPs as described in Table 9. Similar to the earlier use case the validation accuracy of many networks is 100 % but selecting a network that generalizes well to the test set is challenging and still an open question [54]. We can examine the network performance by visualizing the embedded features after the global pooling layer. The first two PCs of the embedded fea-

Table 9: Summary of parameters used for the WaveNet-based network.

Layers	Filter Size (H × W)	Number of filters	Stride	Dilation Factor
Conv Block 1	1 × 50	80	1 × 3	1 × 1
WaveNet Block	1 × 5	80	1 × 1	$5^{(\text{BlockNumber-1})}$
Conv Block 2	1 × 4	80	1 × 2	1 × 1
Pooling 1	1 × 4	–	1 × 4	–
Conv Block 3	1 × 8	80	1 × 1	1 × 1
Conv Block 4	1 × 8	80	1 × 1	1 × 1
Pooling 2	1 × 8	–	1 × 8	–
Final Conv	1 × 1	4	1 × 1	1 × 1

tures are illustrated in Figure 9b, which clearly shows that features of the test data are significantly shifted from the training data for all target classes.

3.2.2 Domain adaptation

Although the test accuracy of the trained FESC stack is almost perfect (98.8±0.8 %), we still use calibration and adjustment to compare the results. For this use case shifts from the target groups are different for each individual class therefore using a single class to calibrate the test set is not sufficient. This is evident in Figure 9a; if we move the test data for the healthy state to the mean value of the training set and then apply the same distance for other classes, it increases the observed shifts for the other classes considerably. One solution is applying standardization using a small portion of the test set from all classes. Thus, 20 % of test data from each class was used for this form of calibration and adjustment. The labels of the recalibration data are not needed. Figure 11 shows the results after standardizing the training and test data for both the FESC stack and the ANN model. Quantitative results are presented in Table 10; because of the stochastic evaluation procedure, the mean and standard deviations of 10 different runs are reported. Similar to the HS use case, a significant improvement is achieved for both ML approaches with the proposed domain adaptation, however, the performance of the FESC approach for the domain shift is significantly better than for the deep network. Furthermore, it had proved to be more robust to the domain shift even before domain adaptation, i. e., might be considered as domain generalization.

4 Conclusion and future works

In this paper DNNs were compared with conventional methods based on feature extraction and selection in scenarios with distribution shifts caused by changing ambient or experimental conditions. By visualizing the data at different levels, it was shown how shifts from raw data

Table 10: Accuracy the models for CWRU dataset, before and after recalibration.

Model	Validation Accuracy %	Test Accuracy %	Test Accuracy % after recalibration
FESC	100	98.8	99.7 ± 0.3
WaveNet-style	100	81	92.5 ± 0.5

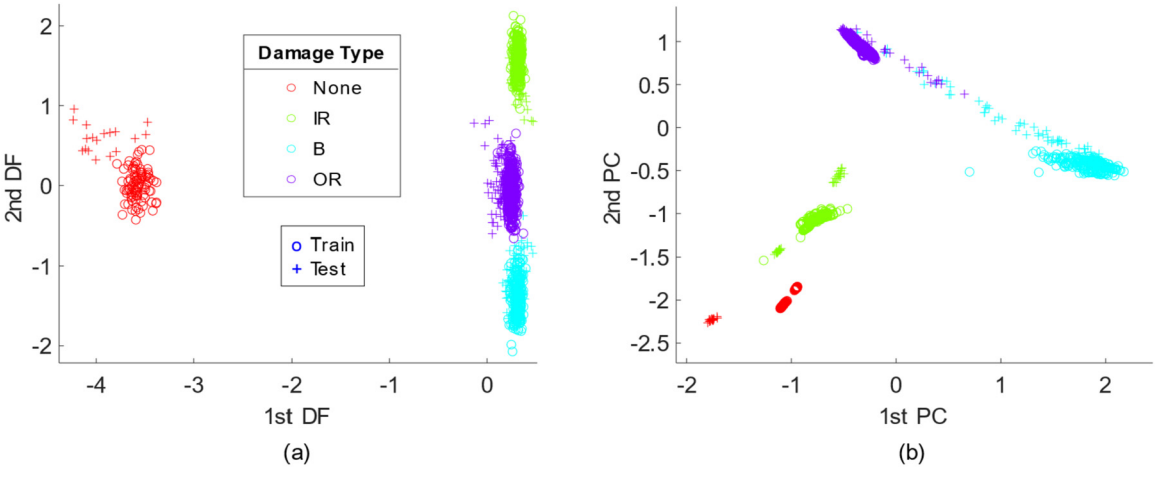


Figure 11: LDA projection of the features in the CWRU use case after recalibration (a). Embedded features from the network in the last convolution layer after recalibration, the graph is the first and second PCs of the features (b).

can propagate to a model and cause shifts in the predictions. As shifts in the data distribution are inevitable in many real-life scenarios, this issue needs to be considered when building a comprehensive ML model, i. e., in the model selection, validation scenario, training process adaptation. In the presented scenarios the conventional FESC/R approaches show better results compared to the ANN solutions. Although finding a DNN to correctly predict the training data is not difficult using NAS algorithms selecting a network that generalizes to the test data is highly challenging in a cross-domain situation. We also presented two simple domain adaptation techniques to improve the results of trained models. This showed that domain adaptation can be formulated as recalibration especially for regression use-cases achieving good results for both ML approaches, but again with significant advantages for the conventional approach. For classification tasks this recalibration is not as straightforward due to the categorical nature of the target data and did not show significant improvement. Again, the conventional approach proved to be more robust against distribution shifts and did achieve better performance after recalibration by normalization. Moreover, in the CWRU use case the FESC method achieves near perfect accuracy for a cross-domain scenario even before recalibration, thus can be considered as an example for domain generalization.

For future work further investigation is suggested in why FESC/R performs better than DNNs in inter-domain scenarios which could help in improving the ANN architectures making them more robust for real-world applications. One could assume that this results from the implicit extraction of useful information from the data during the feature extraction and selection steps reducing the task complexity and making the results more stable with respect to possible changes in the input data. On the other hand, the classical approach can be boosted by explicitly introducing non-linearities based on polynomial expansion of the features in combination with linear classification/regression algorithms as recently suggested [55]. This might allow better adaptation to non-linear dependencies, which is an area where ANNs are usually superior. Also, visualization methods are desirable to indicate where domain shifts occur. Finally, for industrial applications investigating applicable domain adaptation methods and algorithms is necessary because almost all real-life scenarios need to be robust against domain shift situations.

Funding: This work was in part supported by the European Regional Development Fund (Europäischer Fonds für regionale Entwicklung, EFRE) within the project „SE-ProEng“. This work was in part supported by the German ministry for education and research (BMBF) within the project „KI-MUSIK4.0“ under code 16ME0077.

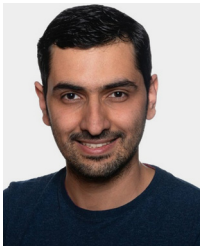
References

1. R. K. Mobley, An introduction to predictive maintenance. Elsevier, 2002.
2. A. Schütze, N. Helwig, and T. Schneider, “Sensors 4.0 – smart sensors and measurement technology enable Industry 4.0,” Journal of Sensors and Sensor systems, vol. 7, no. 1, pp. 359–371, 2018.
3. D. C. Montgomery, Design and analysis of experiments. John Wiley & Sons, 2017.
4. P. W. Koh et al., “WILDS: A Benchmark of in-the-Wild Distribution Shifts,” International Conference on Machine Learning, pp. 5637–5664, Dec. 2021.
5. J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. v. Chawla, and F. Herrera, “A unifying view on dataset shift in classification,” Pattern Recognition, vol. 45, no. 1, pp. 521–530, Jan. 2012, doi: 10.1016/j.patcog.2011.06.019.
6. G. Widmer and M. Kubat, “Learning in the presence of concept drift and hidden contexts,” Machine Learning, vol. 23, no. 1, pp. 69–101, Apr. 1996, doi: 10.1007/BF00116900.
7. M. G. Kelly, D. J. Hand, N. M. Adams, “The impact of changing populations on classifier performance,” Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 367–371, 1999, doi: 10.1145/312129.312285.
8. H. Shimodaira, “Improving predictive inference under covariate shift by weighting the log-likelihood function,” Journal of Statistical Planning and Inference, vol. 90, no. 2, pp. 227–244, Oct. 2000, doi: 10.1016/S0378-3758(00)00115-4.
9. D. A. Cieslak and N. v. Chawla, “A framework for monitoring classifiers’ performance: when and why failure occurs?,” Knowledge and Information Systems, vol. 18, no. 1, pp. 83–108, Jan. 2009, doi: 10.1007/s10115-008-0139-1.
10. R. Alaiz-Rodríguez, A. Guerrero-Curieses, and J. Cid-Sueiro, “Minimax regret classifier for imprecise class distributions,” Journal of Machine Learning Research, vol. 8, pp. 103–130, 2007.
11. R. Caruana, “Multitask Learning,” Machine Learning, vol. 28, no. 1, pp. 41–75, 1997, doi: 10.1023/A:1007379606734.
12. A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, “A Kernel Two-Sample Test,” Journal of Machine Learning Research, vol. 13, no. 25, pp. 723–773, 2012.
13. K. Saenko, B. Kulis, M. Fritz, and T. Darrell, “Adapting Visual Category Models to New Domains,” European conference on computer vision, pp. 213–226, 2010, doi: 10.1007/978-3-642-15561-1_16.
14. Y. Ganin and V. Lempitsky, “Unsupervised Domain Adaptation by Backpropagation,” ICML’15: Proceedings of the 32nd International Conference on Machine Learning – Volume 37, pp. 1180–1189, Jul. 2015.
15. Y. Ganin et al., “Domain-adversarial training of neural networks,” The journal of machine learning research, vol. 17, no. 1, pp. 2030–2096, 2016.
16. E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, “Adversarial

Discriminative Domain Adaptation," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2962–2971, Jul. 2017, doi: 10.1109/CVPR.2017.316.
17. X. Li, W. Zhang, Q. Ding, and J.-Q. Sun, "Multi-Layer domain adaptation method for rolling bearing fault diagnosis," Signal Processing, vol. 157, pp. 180–197, Apr. 2019, doi: 10.1016/j.sigpro.2018.12.005.
18. W.-G. Chang, T. You, S. Seo, S. Kwak, and B. Han, "Domain-Specific Batch Normalization for Unsupervised Domain Adaptation," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2019.
19. Z. Lu, Y. Yang, X. Zhu, C. Liu, Y.-Z. Song, and T. Xiang, "Stochastic Classifiers for Unsupervised Domain Adaptation," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Jun. 2020.
20. W. Li, Z. Yuan, W. Sun, and Y. Liu, "Domain Adaptation for Intelligent Fault Diagnosis under Different Working Conditions," MATEC Web of Conferences, vol. 319, p. 03001, Sep. 2020, doi: 10.1051/matecconf/202031903001.
21. X. Wang, F. Liu, and D. Zhao, "Cross-Machine Fault Diagnosis with Semi-Supervised Discriminative Adversarial Domain Adaptation," Sensors, vol. 20, no. 13, p. 3753, Jul. 2020, doi: 10.3390/s20133753.
22. W. Zhang, G. Peng, C. Li, Y. Chen, and Z. Zhang, "A New Deep Learning Model for Fault Diagnosis with Good Anti-Noise and Domain Adaptation Ability on Raw Vibration Signals," Sensors, vol. 17, no. 2, p. 425, Feb. 2017, doi: 10.3390/s17020425.
23. N. Helwig, E. Pignanelli, and A. Schütze, "Condition monitoring of a complex hydraulic system using multivariate statistics," 2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Proceedings, pp. 210–215, May 2015, doi: 10.1109/I2MTC.2015.7151267.
24. T. Schneider, S. Klein, and M. Bastuck, "Condition monitoring of hydraulic systems Data Set at ZeMA," Zenodo, Apr. 2018, doi: 10.5281/ZENODO.1323611.
25. "Case Western Reserve University Bearing Data Set," Case Western Reserve University Bearing Data Center. https://engineering.case.edu/bearingdatacenter.
26. S. Zhang, S. Zhang, B. Wang, and T. G. Habetler, "Deep learning algorithms for bearing fault diagnostics – A comprehensive review," IEEE Access, vol. 8, pp. 29857–29881, 2020.
27. T. Schneider, N. Helwig, and A. Schütze, "Industrial condition monitoring with smart sensors using automated feature extraction and selection," Measurement Science and Technology, vol. 29, no. 9, p. 94002, Aug. 2018, doi: 10.1088/1361-6501/aad1d4.
28. A. Widodo and B.-S. Yang, "Support vector machine in machine condition monitoring and fault diagnosis," Mechanical Systems and Signal Processing, vol. 21, no. 6, pp. 2560–2574, Aug. 2007, doi: 10.1016/j.ymssp.2006.12.007.
29. D.-T. Hoang and H.-J. Kang, "A survey on Deep Learning based bearing fault diagnosis," Neurocomputing, vol. 335, pp. 327–335, Mar. 2019, doi: 10.1016/j.neucom.2018.06.078.
30. M. S. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, and A. P. Sheth, "Machine learning for internet of things data analysis: a survey," Digital Communications and Networks, vol. 4, no. 3, pp. 161–175, 2018, doi: 10.1016/j.dcan.2017.10.002.
31. W. Zhang, D. Yang, and H. Wang, "Data-Driven Methods for Predictive Maintenance of Industrial Equipment: A Survey," IEEE Systems Journal, vol. 13, no. 3, pp. 2213–2227, Sep. 2019, doi: 10.1109/JSYST.2019.2905565.
32. A. Preece, D. Harborne, D. Braines, R. Tomsett, and S. Chakraborty, "Stakeholders in Explainable AI," arXiv preprint arXiv:1810.00184, Sep. 2018.
33. C. Schorr, P. Goodarzi, F. Chen, and T. Dahmen, "Neuroscope: An Explainable AI Toolbox for Semantic Segmentation and Image Classification of Convolutional Neural Nets," Applied Sciences, vol. 11, no. 5, 2021, doi: 10.3390/app11052199.
34. T. Dorst, Y. Robin, S. Eichstädt, A. Schütze, and T. Schneider, "Influence of synchronization within a sensor network on machine learning results," Journal of Sensors and Sensor Systems, vol. 10, no. 2, pp. 233–245, Aug. 2021, doi: 10.5194/jsss-10-233-2021.
35. Y. Robin, P. Goodarzi, T. Baur, C. Schultealbert, A. Schütze, and T. Schneider, "Machine Learning based calibration time reduction for Gas Sensors in Temperature Cycled Operation," 2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), pp. 1–6, May 2021, doi: 10.1109/I2MTC50364.2021.9459919.
36. T. Schneider, S. Klein, and A. Schütze, "Machine learning in industrial measurement technology for detection of known and unknown faults of equipment and sensors," tm – Technisches Messen, vol. 86, no. 11, pp. 706–718, Nov. 2019, doi: 10.1515/teme-2019-0086.
37. Y. Robin et al., "High-Performance VOC Quantification for IAQ Monitoring Using Advanced Sensor Systems and Deep Learning," Atmosphere, vol. 12, no. 11, p. 1487, Nov. 2021, doi: 10.3390/atmos12111487.
38. I. Kononenko, E. Šimec, and M. Robnik-Šikonja, "Overcoming the Myopia of Inductive Learning Algorithms with RELIEFF," Applied Intelligence, vol. 7, no. 1, pp. 39–55, 1997, doi: 10.1023/A:1008280620621.
39. R. A. Fisher, "The Use of Multiple Measurements in Taxonomic Problems," Annals of Eugenics, vol. 7, no. 2, pp. 179–188, Sep. 1936, doi: 10.1111/j.1469-1809.1936.tb02137.x.
40. S. Wold, M. Sjöström, and L. Eriksson, "PLS-regression: a basic tool of chemometrics," Chemometrics and Intelligent Laboratory Systems, vol. 58, no. 2, pp. 109–130, Oct. 2001, doi: 10.1016/S0169-7439(01)00155-1.
41. T. Brown et al., "Language Models are Few-Shot Learners," Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901. 2020.
42. Z. Dai, H. Liu, Q. v. Le, and M. Tan, "CoAtNet: Marrying Convolution and Attention for All Data Sizes," arXiv preprint arXiv:2106.04803, June 2021.
43. P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, "Sharpness-Aware Minimization for Efficiently Improving Generalization," arXiv preprint arXiv:2010.01412, Oct. 2020.
44. B. Zoph and Q. v. Le, "Neural Architecture Search with Reinforcement Learning," arXiv preprint arXiv:1611.01578, Nov. 2016.
45. T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," The Journal of Machine Learning Research, vol. 20, no. 1, pp. 1997–2017, 2019.
46. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2016.
47. A. van den Oord et al., "WaveNet: A Generative Model for Raw Audio," arXiv preprint arXiv:1609.03499, Sep. 2016.

48. J. Snoek, H. Larochelle, and R. P. Adams, “Practical Bayesian Optimization of Machine Learning Algorithms,” Advances in neural information processing systems, vol. 25, Jun. 2012.
49. M. Holschneider, R. Kronland-Martinet, J. Morlet, and Ph. Tchamitchian, “A Real-Time Algorithm for Signal Analysis with the Help of the Wavelet Transform,” in Wavelets, Springer, Berlin, Heidelberg, 1990, pp. 286–297, doi: 10.1007/978-3-642-75988-8_28.
50. C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, “Understanding deep learning (still) requires rethinking generalization,” Communications of the ACM, vol. 64, no. 3, pp. 107–115, Mar. 2021, doi: 10.1145/3446776.
51. R. Taori, A. Dave, V. Shankar, N. Carlini, B. Recht, and L. Schmidt, “Measuring Robustness to Natural Distribution Shifts in Image Classification,” arXiv preprint arXiv:2007.00644, Jul. 2020.
52. D. N. Perkins, G. Salomon, et al., “Transfer of learning,” International encyclopedia of education, vol. 2, pp. 6452–6457, 1992.
53. S. Bozinovski, “Reminder of the First Paper on Transfer Learning in Neural Networks, 1976,” Informatica (Slovenia), vol. 44, 2020.
54. I. Gulrajani and D. Lopez-Paz, “In Search of Lost Domain Generalization,” arXiv preprint arXiv:2007.01434, Jul. 2020.
55. S. Youssef, “Einsatz maschineller Lernalgorithmen zur mikromagnetischen Materialcharakterisierung,” dissertation, Saarland University, 2021.

Bionotes



Payman Goodarzi
Universität des Saarlandes, Lab for Measurement Technology, 66123 Saarbrücken, Germany
p.goodarzi@lmt.uni-saarland.de

Payman Goodarzi studied Embedded Systems at Saarland University and received his Master of Science degree in March 2020 with a thesis on the interpretability of neural networks. Since that time, he has been working at the Lab for Measurement Technology (LMT) of Saarland University and at the Centre for Mechatronics and Automation Technology (ZeMA) as a scientific researcher. His research interests include ML and deep learning for condition monitoring of technical systems.

Andreas Schütze
Universität des Saarlandes, Lab for Measurement Technology, 66123 Saarbrücken, Germany
schuetze@lmt.uni-saarland.de

Andreas Schütze received his diploma in physics from RWTH Aachen in 1990 and his doctorate in Applied Physics from Justus-Liebig-Universität in Gießen in 1994 with a thesis on microsensors and sensor systems for the detection of reducing and oxidizing gases. From 1994 until 1998 he worked for VDI/VDE-IT, Teltow, Germany, mainly in the fields of microsystems technology. From 1998 until 2000 he was professor for Sensors and Microsystem Technology at the University of Applied Sciences in Krefeld, Germany. Since April 2000 he is professor for Measurement Technology in the Department Systems Engineering at Saarland University, Saarbrücken, Germany and head of the Laboratory for Measurement Technology (LMT). His research interests include smart gas sensor systems as well as data engineering methods for industrial applications.

Tizian Schneider
Universität des Saarlandes, Lab for Measurement Technology, 66123 Saarbrücken, Germany
t.schneider@lmt.uni-saarland.de

Tizian Schneider studied Microtechnologies and Nanostructures at Saarland University and received his Master of Science degree in January 2016. Since that time, he has been working at the Lab for Measurement Technology (LMT) of Saarland University and at the Centre for Mechatronics and Automation Technology (ZeMA) leading the research group Data Engineering & Smart Sensors. His research interests include ML methods for condition monitoring of technical systems, automatic ML model building and interpretable AI.

6.2 Paper C: Uncertainty-aware Automated Machine Learning Toolbox

Consequent error propagation through the toolbox allows comparing errors caused by the sensor's measurement uncertainty to systematic errors caused by domain shifts (assessable via leave one group out cross-validation). It also enables traceability of measurement values. This error propagation was researched in the following paper. The resulting analytical formulas were published in addition to the automated machine learning toolbox, saving significant computational costs compared to alternative Monte Carlo simulations. Additionally, they allow the user to consider uncertainty in feature selection, as shown by the suggested uncertainty-weighted Pearson correlation coefficient for feature ranking.

Uncertainty-aware Automated Machine Learning Toolbox

Tanja Dorst[1], Tizian Schneider[1], Sascha Eichstädt[2], and Andreas Schütze[3]

[1]*Centre for Mechatronics and Automation Technology (ZeMA gGmbH), Saarbrücken, Germany*

[2]*Physikalisch-Technische Bundesanstalt PTB, Braunschweig, Germany*

[3]*Saarland University, Lab for Measurement Technology, Saarbrücken, Germany*

tm – Technisches Messen (2023), 90 (3), 141-153

The original paper can be found online at https://doi.org/10.1515/teme-2022-004.

© Used with permission of Walter de Gruyter and Company, from Uncertainty-aware Automated Machine Learning Toolbox, Dorst, Tanja; Schneider, Tizian; Eichstädt, Sascha; Schütze, Andreas, 90, 3, 2023; permission conveyed through Copyright Clearance Center, Inc.

Tanja Dorst*, Tizian Schneider, Sascha Eichstädt, and Andreas Schütze

Uncertainty-aware automated machine learning toolbox

Automatisierte Toolbox für maschinelles Lernen unter Berücksichtigung von Messunsicherheiten

https://doi.org/10.1515/teme-2022-0042
Received March 29, 2022; accepted September 13, 2022

Abstract: Measurement data can be considered complete only with an associated measurement uncertainty to express knowledge about the spread of values reasonably attributed to the measurand. Measurement uncertainty also allows to assess the comparability and the reliability of measurement results as well as to evaluate decisions based on the measurement result. Artificial Intelligence (AI) methods and especially Machine Learning (ML) are often based on measurements, but so far, uncertainty is widely neglected in this field. We propose to apply uncertainty propagation in ML to allow estimating the uncertainty of ML results and, furthermore, an optimization of ML methods to minimize this uncertainty. Here, we present an extension of a previously published automated ML toolbox (AMLT), which performs feature extraction, feature selection and classification in an automated way without any expert knowledge. To this end, we propose to apply the principles described in the "Guide to the Expression of Uncertainty in Measurement" (GUM) and its supplements to carry out uncertainty propagation for every step in the AMLT. In previous publications we have presented the uncertainty propagation for some of the feature extraction methods in the AMLT. In this contribution, we add some more elements to this concept by also including statistical moments as a feature extraction method, add uncertainty propagation to the feature selection methods and extend it to also include the classification method, linear discriminant analysis combined with Mahalanobis distance. For these methods, analytical approaches for uncertainty propagation are derived in detail, and the uncertainty propagation for the other feature extraction and selection methods are briefly revisited. Finally, the use the uncertainty-aware AMLT is demonstrated for a data set consisting of uncorrelated measurement data and associated uncertainties.

Keywords: Measurement uncertainty, uncertainty propagation, statistical moments, linear discriminant analysis, machine learning.

Zusammenfassung: Messdaten können nur dann als vollständig angesehen werden, wenn sie mit einer Messunsicherheit versehen sind, die das Wissen über die Streuung der Werte ausdrückt, die der Messgröße zugeordnet werden kann. Die Messunsicherheit ermöglicht zudem die Beurteilung der Vergleichbarkeit und Zuverlässigkeit von Messergebnissen sowie die Bewertung von Entscheidungen auf der Grundlage von Messergebnissen. Methoden der künstlichen Intelligenz (KI) und insbesondere des maschinellen Lernens (ML) basieren häufig auf Messungen, aber bisher wurde die Unsicherheit in diesem Bereich weitgehend vernachlässigt. Wir schlagen daher in diesem Beitrag vor, die Unsicherheitsfortpflanzung beim ML anzuwenden, um die Unsicherheit von ML-Ergebnissen abzuschätzen und darüber hinaus eine Optimierung von ML-Methoden zur Minimierung dieser Unsicherheit zu ermöglichen. Dazu stellen wir eine Erweiterung einer bereits veröffentlichten automatisierten ML-Toolbox (AMLT) vor, die Merkmalsextraktion, Merkmalsselektion und Klassifikation automatisiert und ohne Expertenwissen durchführt. Die im „Guide to the Expression of Uncertainty in Measurement" (GUM) und seinen Supplementen beschriebenen Prinzipien werden angewandt, um eine Unsicherheitsfortpflanzung für jeden Schritt in der AMLT durchzuführen. In früheren Veröffentlichungen haben wir bereits die Unsicherheitsfortpflanzung für einige der Merkmalsextraktionsmethoden in der AMLT vorgestellt. In diesem Beitrag fügen wir nun diesem Konzept einige weitere Elemente hinzu, indem wir auch statistische Momente als Merkmalsextraktionsmethode einbeziehen, die Unsi-

***Corresponding author: Tanja Dorst,** ZeMA – Center for Mechatronics and Automation Technology gGmbH, Saarbrücken, Germany, e-mail: t.dorst@zema.de, ORCID: https://orcid.org/0000-0001-9756-9014
Tizian Schneider, Andreas Schütze, ZeMA – Center for Mechatronics and Automation Technology gGmbH, Saarbrücken, Germany; and Lab for Measurement Technology, Department of Mechatronics, Saarland University, Saarbrücken, Germany, ORCID: https://orcid.org/0000-0003-3060-5177 (A. Schütze)
Sascha Eichstädt, Physikalisch-Technische Bundesanstalt, Braunschweig and Berlin, Germany, ORCID: https://orcid.org/0000-0001-7433-583X

Published online September 22, 2022

cherheitsfortpflanzung zu den Merkmalselektionsmethoden hinzufügen und sie auch auf die Klassifikationsmethode, die lineare Diskriminanzanalyse in Kombination mit der Mahalanobis-Distanz, ausweiten. Für diese Methoden werden analytische Ansätze für die Unsicherheitsfortpflanzung im Detail abgeleitet, und die Unsicherheitsfortpflanzungen für die anderen Merkmalsextraktions- und -selektionsmethoden werden kurz aufgegriffen. Abschließend wird die Anwendung der zuvor vorgestellten Version der AMLT, welche Unsicherheiten berücksichtig, für einen Datensatz, welcher aus unkorrelierten Messdaten und dazugehörigen Unsicherheiten besteht, demonstriert.

Schlagwörter: Messunsicherheit, Unsicherheitsfortpflanzung, statistische Momente, lineare Diskriminanzanalyse, maschinelles Lernen.

1 Introduction

Whenever decisions are based on machine learning (ML) inference, it is important to have an assessment of the reliability of the ML results. This reliability is very much affected by the quality of the input data, e. g., the measurements. Measurement uncertainties, calibration, and traceability of measurements to the International System of Units (SI) belong to the most important basic metrological principles.

In [1] and [2], an automated software toolbox for statistical ML was presented. It is suited for multi-class classification problems using cyclic sensor data which means that every cycle must have the same length or continuous data must be split into cycles of same length. Cycles are classified to exactly one class. In this contribution, this automated ML toolbox (AMLT) is extended by consideration of measurement uncertainty. The mathematical focus is especially on two different methods and their corresponding uncertainty propagation: Statistical moments as feature extraction and Linear Discriminant Analysis (LDA) as dimensionality reduction method. To complete the uncertainty-aware AMLT, the uncertainty propagation for the other feature extraction and selection methods are briefly revisited.

With the help of statistical moments, characteristics of the statistical distribution of measurement values can be described and used as features. In pattern recognition, LDA is used as a linear dimensionality reduction technique to achieve a more manageable number of features before the actual classification and to reduce the computational cost. Existing classical statistical methods for dimensionality reduction have been developed in a time period, when data collection and storage was not as readily available as it is today, and the size of the data sets was much smaller. In 1936, Fisher introduced LDA on the example of the well-known multivariate Fisher's Iris data set [3]. LDA is a method for finding linear combinations of variables that separate observations into two or more classes by minimizing the ratio of intra-class to inter-class variance. Nowadays, in the era of big data, massive amounts of data are generated in various application domains worldwide, leading (in particular) to an increase in dimensionality and data size [4]. Computations in high dimensional spaces can lead to overfitting [5] or the curse of dimensionality [6, 7] as high dimensional spaces have counterintuitive geometrical properties.

To be capable of evaluating the data quality and therefore the quality of the machine learning results within the framework of a measurement uncertainty analysis, using data and its associated measurement uncertainty is necessary. The easiest way to determine measurement uncertainty is to use calibration information, e. g., from a calibration certificate, but a calibration is costly and therefore often not performed. In the case of existing assembly lines and test beds, it could also be difficult or impossible to dismount process-critical sensors and subsequently recalibrate them. In case no calibration information is available, uncertainty information provided by the manufacturers of the sensors in data sheets can be used to obtain an indication of the data quality in the form of a measurement uncertainty [8, 9]. In both cases, an uncertainty value can be provided for every measured sensor value. This fulfills the requirements for the use of the uncertainty-aware AMLT presented in this contribution.

2 Automated ML toolbox

To use the AMLT without any expert knowledge in a fully automated way, a data matrix $\mathbf{D} \in \mathbb{R}^{m \times n}$ for each sensor must be given. For cyclic sensor data, this means that the matrix consists of m cycles where each cycle has the same length of n data points. For non-cyclic sensor data, windowing approaches must be performed before getting the data in the format of the data matrix $\mathbf{D}$. The AMLT is divided into three main parts (cf. Fig. 1): feature extraction (FE), feature selection (FS) and classification. In the end, to verify the trained model, a validation is performed.

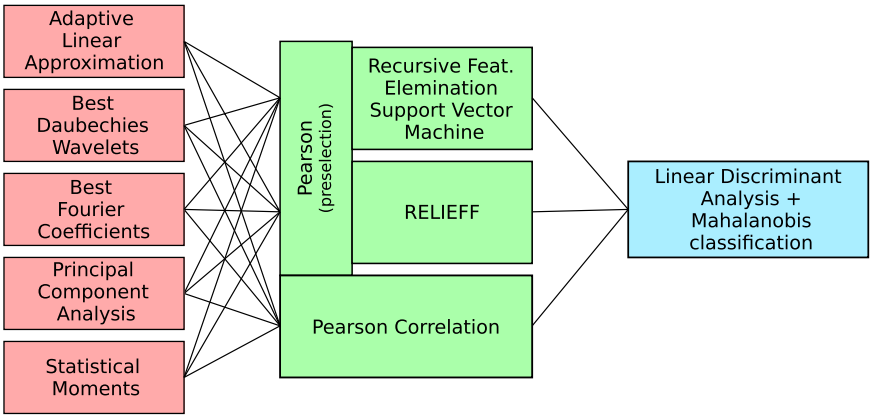


Figure 1: Scheme of the automated ML toolbox (AMLT) with feature extraction (red), feature selection (green) and classification (blue) (adapted from [2]).

2.1 Feature extraction

The objective of the unsupervised FE is to concentrate as much information in as few features as possible. In this step of the AMLT, features are extracted from cyclic raw data $\mathbf{D}$ in different domains by five complementary algorithms:

- Adaptive Linear Approximation (ALA):
  Cycles are split into approximately linear segments. Mean value and slope of every linear segment are extracted as features from time domain [10].
- Best Daubechies Wavelet (BDW):
  A Daubechies D4 (four wavelet and scaling function coefficients) wavelet transform is performed [11]. 10 % of the Wavelet coefficients with the highest average absolute value over all cycles are extracted as features from time-frequency domain.
- Best Fourier Coefficients (BFC):
  10 % of amplitudes with the highest average absolute value over all cycles and their corresponding phases are extracted as features from frequency domain [12].
- Principal Component Analysis (PCA):
  PCA reduces the number of variables of a data set, while preserving as much information as possible [13, 14, 15, 16]. The projections on the first principal components are used as features from time domain.
- Statistical Moments:
  The statistical distribution of the measurement values also includes information. The cycles are divided into $s = 10$ nearly equally sized segments and the four moments mean, standard deviation (as the root of the variance), skewness, and kurtosis are extracted for each segment as features from time domain, resulting in $4s$ features per cycle [17].

Using these algorithms leads to five feature sets with a large number of features included in each one. For each of the five complementary algorithms, FE can be defined as a mapping $\mathbf{D} \mapsto \mathbf{F_E}$, where $\mathbf{F_E} \in \mathbb{R}^{m \times k}$, $k < n$, denotes the matrix containing extracted features. As the data reduction is insufficient for Big Data applications in this step, the number of features is further reduced in the FS step.

2.2 Feature selection

In the supervised FS step, features with low information content and redundant features are removed from each feature set F_E and the most relevant features with respect to the given classification task are selected. Supervised means that the target value, i. e., the associated class, is known. In the AMLT, three complementary algorithms are used for FS.

- Pearson Correlation:
  Due to low computational cost, this algorithm is used for FS itself and for the first preselection step in FS, if the feature number is more than 500 per feature set $\mathbf{F_E}$. Features are arranged in a descending order according to their absolute correlation coefficient. In general, the coefficient in $[-1, 1]$ indicates the strength and direction (in case it is not the absolute value) of the linear relationship between a feature and a target value. A correlation close to 0 indicates no linear relationship.
- Recursive Feature Elimination Support Vector Machine (RFESVM):
  With a linear SVM, an optimal hyperplane with a maximum margin (distance between the hyperplane and

the support vectors) is calculated by solving the optimization problem

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \quad i = 1, \ldots, l. \end{aligned} \tag{1}$$

In this equation, $\mathbf{w}$ is a weight vector and b is a scalar, called bias. $\mathbf{x}_i$ are the support vectors and y_i the labels which are ± 1 for binary classification problems. The lowest SVM weights $\mathbf{w}$ are used to recursively remove the features with lowest contribution to the group separation from the feature set $\mathbf{F_E}$ [18, 19]. For multi-class classification, One-vs-One is used that splits the multi-class into binary classification problems, i. e., one for every possible pair of classes, and the results are averaged.

– ReliefF:
  In case of an impossible linear group separation, ReliefF is used which denotes the sixth algorithm version (naming from A to F) of Relief [20, 21]. ReliefF deals with multi-class problems. It finds the nearest hits and nearest misses for each point by using k-nearest neighbors with the Manhattan metric (induced by 1-norm) as distance measure [22, 23, 24]. For one point, this means that this algorithm identifies several nearest neighbors, one belonging to the same class (nearest hit) and the others each belonging to different classes (nearest misses).

After ranking the features according to the FS algorithms, the following optimization problem is solved. For every number of features, a 10-fold cross-validation (explained in Section 2.4) is carried out and the minimum number l of features with the lowest cross-validation error is determined. Thus, FS can be defined as a mapping $\mathbf{F_E} \mapsto \mathbf{F_S}$, where $\mathbf{F_S} \in \mathbb{R}^{m \times l}$, $l < k$, denotes the matrix containing only the optimum number of the most relevant features.

2.3 Classification

The classification step is divided into two parts. First, there is a further dimensionality reduction performed by LDA and then, the classification itself by using the Mahalanobis distance. In general, the dimensionality reduction does not only reduce computational costs for a given classification task, but it can also avoid overfitting. For g groups, LDA performs a linear projection of the feature space into a smaller $\tilde{g} = g - 1$ dimensional subspace by maximizing the inter-class variance and minimizing the intra-class variance [25]. This results in a projection matrix $\mathbf{P} \in \mathbb{R}^{l \times \tilde{g}}$, where l denotes the optimal number of features and $\tilde{g}$ the number of separable groups reduced by one.

The actual classification task is carried out by using the Mahalanobis distance which measures distances relative to central point of each group [26, 27, 28]. Let $\mathbf{x}$ be the vector with the features of the test data, $\mathbf{m}$ the componentwise arithmetic mean of the features of the training data and $\mathbf{S}$ the covariance matrix of the features of the training data all appertaining to the class C_i. Then, the Mahalanobis distance is defined as

$$d_{\text{Mahal}}(\mathbf{x}, C_i) = \sqrt{(\mathbf{x} - \mathbf{m}_i)^\top \mathbf{S}_i^{-1} (\mathbf{x} - \mathbf{m}_i)}. \tag{2}$$

The class that results of the lowest Mahalanobis distance is assigned to $\mathbf{x}$.

2.4 Validation

To validate the results, a k-fold stratified cross-validation [29] with $k = 10$ is automatically performed by the AMLT. This method equally partitioned the data set into ten subsets where each of the subsets has nearly the same class distribution as the complete data set. The model is trained with only 90 % of the data set (i. e., the training data), then the trained model is applied to the remaining 10 % of the data set (i. e. the test data) and the cross-validation (CV) error, i. e., the percentage of misclassified cycles, is calculated. After performing training, testing and calculation of the CV error for every fold, the calculated CV error values are averaged over all folds and the algorithm combination with lowest averaged CV error is chosen as the best for the actual classification task.

3 Extension of the automated ML toolbox

The extension of the AMLT by consideration of measurement uncertainty is based on the *Guide to the Expression of Uncertainty in Measurement* (GUM) [30] and its supplements *Supplement 1* [31] and *Supplement 2* [32]. The three documents establish general rules for evaluating and expressing measurement uncertainty. In the GUM, the calculation of the measurement uncertainty consists of four main steps:

1. Specification of a measurand.
2. Identification and characterization of the quantities which influence the measurement and evaluation of the uncertainty for each of these influencing quantities.

3. Provision of a mathematical model for the calculation of the measurand, which relates the values of the influencing quantities to the value of the measurand.
4. Calculation of the combined standard measurement uncertainty which is assigned to the measurement result (more precisely the estimated value of the measurand).

In the GUM, a linearization of the model equation $y = f(x_1, x_2, \ldots, x_N)$ is used to combine the individual standard uncertainties according to the Gaussian error propagation (GEP) law

$$u_c^2(y) = \sum_{i=1}^{N}\left(\frac{\partial f}{\partial x_i}\right)^2 u^2(x_i) + \underbrace{2\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\frac{\partial f}{\partial x_i}\frac{\partial f}{\partial x_j}u(x_i, x_j)}_{=0, \text{ if uncorrelated input quantities}} \quad (3)$$

which the GUM refers to as "Law of Propagation of Uncertainty" (LPU). Equation (3) is based on a first order Taylor series approximation and the partial derivatives are called sensitivity coefficients. In Supplement 1 of the GUM, this approach of propagation of uncertainties is replaced by a propagation of probability distributions based on a Monte Carlo method, which does not require linearization of the model. Supplement 2 of the GUM defines the linearization method and the Monte Carlo method for multivariate and complex-valued quantities.

3.1 Uncertainty-aware feature extraction

Let the mapping $\mathbf{D} \mapsto \mathbf{F_E}$ with $\mathbf{D} \in \mathbb{R}^{m\times n}$ and $\mathbf{F_E} \in \mathbb{R}^{m\times k}$, $k \leq n$, be given as described above. Knowledge about the uncertainty matrix $\mathbf{U} \in \mathbb{R}^{m\times n}$, which assigns an uncertainty value u_{ij} to a measurement value d_{ij} $\forall i, j$, assumed to be available which means that correlation between different time instants is neglected. Then, the sensitivity coefficients of the mapping $\mathbf{D} \mapsto \mathbf{F_E}$ can be calculated according to the rules established in the GUM and its supplements. This means, that for every feature in $\mathbf{F_E}$, an associated uncertainty value can be derived according to Eq. (3) or a Monte Carlo method which leads to the feature uncertainty matrix $\mathbf{U}_{\mathbf{F_E}}$. In this contribution, all covariances between feature uncertainties are disregarded.

For PCA, an efficient implementation of a Monte Carlo method for uncertainty evaluation is used [33] as an analytical approach according to Eq. (3) causes numerical issues for Big Data. However, these analytical approaches are applied for all other FE methods included in the AMLT. For ALA, the derivatives of mean and slope for every linear segment are calculated and used as sensitivity coefficients [34, 35]. The derivatives of the real and imaginary part of the discrete Fourier transform are used to calculate the sensitivity coefficients for the amplitude/phase representation in the BFC algorithm [36]. An uncertainty-aware BDW was proposed in [37, 38, 39] and adapted to Daubechies D4 wavelet in [40].

As the uncertainty propagation for statistical moments in line with the GUM has not been published before, the formulas for applying GUM to this algorithm of the FE step are given in brief in this contribution. Using statistical moments as FE algorithm, the cycles are divided into s segments. The start index a_p and the end index e_p of the p-th segment is given by

$$a_p = (p-1)\cdot\left\lceil\frac{n}{s}\right\rceil + 1 \text{ and} \quad (4)$$

$$e_p = \min\left(n, p\cdot\left\lceil\frac{n}{s}\right\rceil\right), \quad (5)$$

such that every segment consists of $N_p = e_p - a_p + 1$ measurement values. For the p-th segment of one cycle (consisting of $d_j \in \{d_{a_p}, \ldots, d_{e_p}\}$), the four statistical moments and their associated sensitivity coefficients are derived as follows, whereas detailed calculations of the formulas can be found in Appendices A.1 to A.3.

– The mean value is calculated by

$$\mu_p = \overline{d_p} = \frac{1}{N_p}\sum_{j=a_p}^{e_p} d_j. \quad (6)$$

As it can be easily seen, the sensitivity coefficients are given by

$$\alpha_{p,j} = \frac{\partial \mu_p}{\partial d_j} = \frac{1}{N_p}. \quad (7)$$

– The standard deviation can be written as

$$\sigma_p = \sqrt{\frac{1}{N_p - 1}\sum_{j=a_p}^{e_p}(d_j - \overline{d_p})^2}. \quad (8)$$

The sensitivity coefficients are calculated with

$$\beta_{p,j} = \frac{\partial \sigma_p}{\partial d_j} = \frac{d_j - \overline{d_p}}{(N_p - 1)\cdot\sigma_p}. \quad (9)$$

– The formula of the skewness is given by

$$\nu_p = \frac{\frac{1}{N_p}\sum_{j=a_p}^{e_p}(d_j - \overline{d_p})^3}{(\frac{1}{N_p}\sum_{j=a_p}^{e_p}(d_j - \overline{d_p})^2)^{\frac{3}{2}}} := \frac{\nu_p^{\text{denom}}}{\nu_p^{\text{nom}}}. \quad (10)$$

To get the sensitivity coefficients, a calculation for the derivatives of the denominator and the nominator of ν_p is performed separately. Then, it holds

$$\frac{\partial v_p^{\text{denom}}}{\partial d_j} = \frac{3}{N_p} \cdot \left((d_j - \overline{d_p})^2 - \frac{1}{N_p} \sum_{j=a_p}^{e_p} (d_j - \overline{d_p})^2 \right) \quad (11)$$

and

$$\frac{\partial v_p^{\text{nom}}}{\partial d_j} = \frac{3}{N_p} \cdot \left(\frac{1}{N_p} \sum_{j=a_p}^{e_p} (d_j - \overline{d_p})^2 \right)^{\frac{1}{2}} \cdot (d_j - \overline{d_p}). \quad (12)$$

Both, Eq. (11) and Eq. (12), together with the quotient rule lead to the sensitivity coefficients $\gamma_{p,j}$.

– Finally, for the kurtosis

$$w_p = \frac{\frac{1}{N_p} \sum_{j=a_p}^{e_p} (d_j - \overline{d_p})^4}{(\frac{1}{N_p} \sum_{j=a_p}^{e_p} (d_j - \overline{d_p})^2)^2} := \frac{w_p^{\text{denom}}}{w_p^{\text{nom}}}, \quad (13)$$

the derivatives of the denominator and nominator of w_p are given by

$$\frac{\partial w_p^{\text{denom}}}{\partial d_j} = \frac{4}{N_p} \cdot \left((d_j - \overline{d_p})^3 - \frac{1}{N_p} \sum_{j=a_p}^{e_p} (d_j - \overline{d_p})^3 \right) \quad (14)$$

and

$$\frac{\partial w_p^{\text{nom}}}{\partial d_j} = \frac{4}{N_p^2} \cdot \left(\sum_{j=a_p}^{e_p} (d_j - \overline{d_p})^2 \right) \cdot (d_j - \overline{d_p}). \quad (15)$$

Inserting Eq. (14) and Eq. (15) in the quotient rule results in the sensitivity coefficients $\delta_{p,j}$.

The sensitivity matrix for every q-th cycle is thus given as a block matrix

$$\mathbf{J}^q_{\alpha,\beta,\gamma,\delta} = \begin{pmatrix} \mathbf{A} \\ \mathbf{B} \\ \mathbf{\Gamma} \\ \mathbf{\Delta} \end{pmatrix} \in \mathbb{R}^{4s \times n} \quad (16)$$

with the submatrices $\mathbf{A} \in \mathbb{R}^{s \times n}$, $\mathbf{B} \in \mathbb{R}^{s \times n}$, $\mathbf{\Gamma} \in \mathbb{R}^{s \times n}$ and $\mathbf{\Delta} \in \mathbb{R}^{s \times n}$. The matrix $\mathbf{J}^q_{\alpha,\beta,\gamma,\delta}$ contains an enormous amount of zeros, e. g., $\alpha_{p,j} = 0$ if $j \notin \{a_p, \ldots, e_p\}$.

Assume that the covariance matrix $\mathbf{U_c} \in \mathbb{R}^{n \times n}$ for every cycle is given. It has the diagonal elements $u_c(d_j, d_j)$ being the squared standard uncertainties $u_c^2(d_j)$ for $j = 1, \ldots, n$ and the off-diagonal elements being the covariances $u_c(d_i, d_j) = u_c(d_i) u_c(d_j) r(d_i, d_j)$ for $i, j = 1, \ldots, n$ and $i \neq j$, where $r(d_i, d_j)$ denotes the correlation coefficient. It holds $r(d_i, d_j) = r(d_j, d_i)$ and $r(d_i, d_j) \in [-1, 1]$. The covariance matrix is symmetric, which means $\mathbf{U_c} = \mathbf{U_c}^\top$. This leads to a symmetric covariance matrix $\mathbf{U} \in \mathbb{R}^{4s \times 4s}$ with

$$\mathbf{U}^q = \mathbf{J}^q_{\alpha,\beta,\gamma,\delta} \cdot \mathbf{U_c} \cdot \left(\mathbf{J}^q_{\alpha,\beta,\gamma,\delta} \right)^\top$$

$$= \begin{pmatrix} \mathbf{A U_c A}^\top & \mathbf{A U_c B}^\top & \mathbf{A U_c \Gamma}^\top & \mathbf{A U_c \Delta}^\top \\ (\mathbf{A U_c B}^\top)^\top & \mathbf{B U_c B}^\top & \mathbf{B U_c \Gamma}^\top & \mathbf{B U_c \Delta}^\top \\ (\mathbf{A U_c \Gamma}^\top)^\top & (\mathbf{B U_c \Gamma}^\top)^\top & \mathbf{\Gamma U_c \Gamma}^\top & \mathbf{\Gamma U_c \Delta}^\top \\ (\mathbf{A U_c \Delta}^\top)^\top & (\mathbf{B U_c \Delta}^\top)^\top & (\mathbf{\Gamma U_c \Delta}^\top)^\top & \mathbf{\Delta U_c \Delta}^\top \end{pmatrix}. \quad (17)$$

As the matrix $\mathbf{U}^q$ is symmetric, it is only necessary to calculate the upper triangle matrix to save computational cost. Detailed information for the matrix multiplication above can be found in Appendix A.4. We assume only white noise in this contribution. The roots of the diagonal entries represent the uncertainty values associated to the features for the q-th cycle and are stored in the q-th row of $\mathbf{U}_{\mathbf{F_E}}$ and the covariances are disregarded. All analytical approaches of uncertainty propagation for the statistical moments were verified by a Monte Carlo simulation. Using the suggested analytical formulas, computational costs can be saved in comparison to the Monte Carlo simulations.

3.2 Uncertainty-aware feature selection

After FE, a feature matrix $\mathbf{F_E} \in \mathbb{R}^{m \times k}$ and the corresponding uncertainty matrix $\mathbf{U}_{\mathbf{F_E}}$ of the same size are available. As FS is a supervised step, the target values $\mathbf{y} \in \mathbb{R}^m$ are known. The uncertainty is further propagated through the different analysis steps including FS. To get the AMLT uncertainty-aware in the FS step, filter methods as weighted rank algorithms are implemented. For weighted Pearson correlation [41], a feature with lower $r_{\text{Pearson},j}$ but small uncertainty is preferred over a feature with higher $r_{\text{Pearson},j}$ but high uncertainty. The weighted Pearson correlation coefficient for feature j with target y is given by

$$r_{\text{Pearson},j} = \frac{\sum_{i=1}^m (w_{ij}(x_{ij} - \overline{x}_j)(y_i - \overline{y}_j))}{[\sum_{i=1}^m (w_{ij}(x_{ij} - \overline{x}_j)^2) \sum_{i=1}^m (w_{ij}(y_i - \overline{y}_j)^2)]^{1/2}}, \quad (18)$$

where w_{ij} denotes a weight for which here the squared reciprocal of the corresponding uncertainty value in $\mathbf{U}_{\mathbf{F_E}}$ is used, $\overline{x}_j$ and $\overline{y}$ are the weighted mean of the j-th column of $\mathbf{F_E}$ and the vector $\mathbf{y}$, respectively, and n is the number of cycles. The Pearson correlation used in the AMLT (cf. Section 2.2) is achieved by assigning w_i the identical weight in Eq. (18). In addition, a weighted Spearman correlation is added to the uncertainty-aware AMLT for use if an at least ordinal scale of the target is used. To get this correlation, all calculations for the values in Eq. (18) are performed for tied ranks [42, 43]. In general, Spearman correlation is used to measure the strength of a monotonic relationship between two variables.

As the filter method ReliefF is based on the Manhattan distance, this distance measure is used in a weighted version in the uncertainty-aware AMLT. Thereby, the distance

along every dimension is weighted with the corresponding uncertainty value.

The wrapper method RFESVM uses a standard binary SVM model with a linear kernel in the AMLT. A total support vector classification (TSVC) is implemented to extend the standard SVM [44]. This support vector classification for uncertain input data is based on a total least squares regression [45]. The noise is given by $\Delta\mathbf{x}_i = \mathbf{x}_i - \mathbf{x}_i'$ in this algorithm, where $\mathbf{x}_i$ denotes a vector with noise and $\mathbf{x}_i'$ one without noise, respectively. A bounded uncertainty noise model $\|\Delta\mathbf{x}_i\| \leq \delta_i$ with uniform prior is assumed. This leads to the following optimization problem [44]:

$$\begin{aligned} \min_{\mathbf{w},b,\Delta\mathbf{x}_i} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to} \quad & y_i(\mathbf{w}^\top(\mathbf{x}_i + \Delta\mathbf{x}_i) + b) \geq 1, \\ & \|\Delta\mathbf{x}_i\| \leq \delta_i, \quad i = 1,\ldots,l. \end{aligned} \tag{19}$$

After performing a TSVC, features with the lowest contribution (weight) to the class separation are then recursively eliminated.

Performing the uncertainty-aware FS yields a feature matrix $\mathbf{F_S} \in \mathbb{R}^{m\times l}$ and the associated uncertainty matrix $\mathbf{U_{F_S}}$ of the same size.

3.3 Uncertainty-aware classification

Let a projection matrix $\mathbf{P} \in \mathbb{R}^{l\times\tilde{g}}$ be given, where l denotes the optimum number of features and $\tilde{g}$ is the number of separable groups reduced by one. $\mathbf{P}$ is calculated during model training without any uncertainty consideration. The matrix of the selected features is given by $\mathbf{F_S} \in \mathbb{R}^{m\times l}$, where m denotes the number of cycles.

3.3.1 Uncertainty-aware LDA

For the LDA transform, it holds

$$\mathbf{L} = \mathbf{F_S}\cdot\mathbf{P} \quad \text{with} \quad \mathbf{L} \in \mathbb{R}^{m\times\tilde{g}}. \tag{20}$$

The calculation of the uncertainty values for $\mathbf{L}$ is based on the formulas given in section 6.2 ("Propagation of uncertainty for explicit multivariate measurement models") of Supplement 2 of the GUM [32]. First, Eq. (20) must be transposed, which leads to

$$\mathbf{L}^\top = \mathbf{P}^\top\cdot\mathbf{F_S}^\top \tag{21}$$

and $\mathbf{F_S}$ and $\mathbf{P}$ must be transformed in a matrix-vector notation. For the columns of $\mathbf{F_S}^\top$, it holds

$$\mathbf{F_S}^\top = (f_1^\top|f_2^\top|\ldots|f_m^\top), \tag{22}$$

where $f_j^\top \in \mathbb{R}^{l\times 1}$, $\forall j = 1,\ldots,m$ denotes the features for the j-th cycle. Thus, the matrix-vector representation is given by

$$\tilde{\mathbf{F}}_\mathbf{S}{}^\top = \begin{pmatrix} f_1^\top \\ f_2^\top \\ \vdots \\ f_m^\top \end{pmatrix} \in \mathbb{R}^{(m\cdot l)\times 1} \tag{23}$$

and

$$\tilde{\mathbf{P}}^\top = \begin{pmatrix} \mathbf{P}^\top & 0 & 0 & \ldots & 0 \\ 0 & \mathbf{P}^\top & 0 & \ldots & 0 \\ 0 & 0 & \mathbf{P}^\top & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & & \ldots & & \mathbf{P}^\top \end{pmatrix} \in \mathbb{R}^{(m\cdot\tilde{g})\times(m\cdot l)}, \tag{24}$$

so that the LDA transform can be expressed by

$$\tilde{\mathbf{L}}^\top = \tilde{\mathbf{P}}^\top\cdot\tilde{\mathbf{F}}_\mathbf{S}{}^\top, \quad \tilde{\mathbf{L}}^\top \in \mathbb{R}^{(m\cdot\tilde{g})\times 1}. \tag{25}$$

Further, let an uncertainty matrix $\mathbf{U_{F_S}}$ of the selected features be given by

$$\mathbf{U_{F_S}} = \begin{pmatrix} u_{11} & u_{12} & \ldots & u_{1l} \\ u_{21} & u_{22} & \ldots & u_{2l} \\ \vdots & \vdots & \vdots & \vdots \\ u_{m1} & u_{m2} & \ldots & u_{ml} \end{pmatrix} \in \mathbb{R}^{m\times l}, \tag{26}$$

where every feature in $\mathbf{F_S}$ the corresponding uncertainty value of $\mathbf{U_{F_S}}$ is associated. The transpose matrix $\mathbf{U_{F_S}}^\top$ is transferred to the diagonal matrix

$$\tilde{\mathbf{U}}_{\mathbf{F_S}}{}^\top = \begin{pmatrix} u_{11} & 0 & 0 & \ldots & 0 \\ 0 & u_{12} & 0 & \ldots & 0 \\ 0 & 0 & u_{13} & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & & \ldots & & u_{ml} \end{pmatrix} \in \mathbb{R}^{(m\cdot l)\times(m\cdot l)}. \tag{27}$$

Using section 6.2.1.3 of [32] leads to the following expression for the covariance matrix $\tilde{\mathbf{U}}$ of $\mathbf{L}$

$$\tilde{\mathbf{U}} = \tilde{\mathbf{P}}^\top\cdot\left(\tilde{\mathbf{U}}_{\mathbf{F_S}}{}^\top\right)^2\cdot\left(\tilde{\mathbf{P}}^\top\right)^\top \tag{28}$$

$$= \tilde{\mathbf{P}}^\top\cdot\left(\tilde{\mathbf{U}}_{\mathbf{F_S}}{}^\top\right)^2\cdot\tilde{\mathbf{P}} \tag{29}$$

with $\tilde{\mathbf{U}} \in \mathbb{R}^{(m\cdot\tilde{g})\times(m\cdot\tilde{g})}$. As there is only an interest for the diagonal elements of $\tilde{\mathbf{U}}$, the formula for calculating the uncertainty values can be simplified and retransformed to

$$\mathbf{U_{LDA}}^{\top} = (\mathbf{P}^{\top} \circ \mathbf{P}^{\top}) \cdot (\mathbf{U_{F_S}}^{\top} \circ \mathbf{U_{F_S}}^{\top}) \tag{30}$$

$$\Leftrightarrow \quad \mathbf{U_{LDA}} = (\mathbf{U_{F_S}} \circ \mathbf{U_{F_S}}) \cdot (\mathbf{P} \circ \mathbf{P}) \tag{31}$$

$$= \mathbf{U_{F_S}}^{\circ 2} \cdot \mathbf{P}^{\circ 2}, \tag{32}$$

where $\circ$ denotes the Hadamard (element-wise) product [46]. The uncertainty values associated with $\mathbf{L}$ can be calculated by

$$\mathbf{U_L} = \left(\left|\mathbf{U_{F_S}}^{\circ 2} \cdot \mathbf{P}^{\circ 2}\right|\right)^{\circ 1/2} \in \mathbb{R}^{m \times \tilde{g}}, \tag{33}$$

where $|.|$ denotes the element-wise absolute value and $(.)^{\circ 1/2}$ the Hadamard (element-wise) square root [47].

3.3.2 Uncertainty-aware Mahalanobis distance classification

Let the matrix of the projected points $\mathbf{L} \in \mathbb{R}^{m \times \tilde{g}}$ and the associated uncertainty matrix $\mathbf{U_L}$ of the same size be given. One projected point is expressed by one row in $\mathbf{L}$ and the associated uncertainty is available in the corresponding row in $\mathbf{U_L}$. For a worst case classification, only points that have the maximum possible distance from a projected point under consideration of the uncertainty values are relevant. In other words, the edges of a hyperrectangle (in total $2^{\tilde{g}}$) are the relevant points which can be calculated by an addition/subtraction of an uncertainty value to the corresponding entry of $\mathbf{L}$. For example, let $\tilde{g} = 3$ be given, so the three-dimensional space is considered. The resulting 2^3 points are the vertices of a cuboid. To perform a classification, Eq. (2) is applied. It calculates the distance between the center of every group and all possible point combinations in the $\tilde{g}$-dimensional space. For every point, the minimum Mahalanobis distance and the corresponding group is determined. In case the uncertainty has no influence on the classification, all points were assigned to the same group. If there is an influence and one or several points are assigned to other groups, this information is available in the prediction graph of the AMLT.

3.4 Application of the uncertainty-aware automated ML toolbox

For the application of the uncertainty-aware AMLT in this contribution, an hydraulic data set is used [48]. In an hydraulic system, different fault conditions of cooler, valve, pump, and accumulator are simulated and data from $m = 1449$ working cycles is recorded using 17 different sensors [49, 50]. The four different fault conditions at various levels of severity are systematically combined, so that the data set contains cycles with each combination of fault conditions. For the exemplary application of the uncertainty-aware AMLT in this contribution, only data of the pressure sensor PS1 and the cooler condition as target is chosen. The hydraulic system operates during the working cycles with cooler conditions of 3 % (close to total failure), 20 % (reduced efficiency), and 100 % (full efficiency). Thus, $g = 3$ separate classes are included in the data set. The sampling rate of PS1 is 100 Hz leading to $n = 6000$ for the machine's 60 s working cycle. As uncertainty contribution for the measured signal, white noise with standard deviation $\sigma = 1$ bar (= 1 kPa) is considered. To use the AMLT for training and application, the data set is divided into training data (90 % corresponding to 1305 cycles) and test data (10 % corresponding to 144 cycles). With statistical moments as FE and the weighted Pearson correlation as FS, the optimum number of features is determined as $l = 27$ by cross-validation on the training data. After the training of the model, the trained model is applied to the test data.

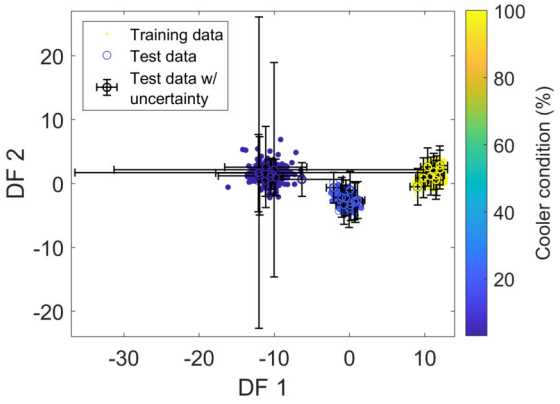


Figure 2: Uncertainty-aware LDA plot for training and test data. Uncertainty is presented as error bar only for every 5$^{\text{th}}$ test data point for better visibility.

Figure 2 shows a two-dimensional LDA plot. For better visibility, only every fifth test data point is depicted with error bars in two directions which indicate the uncertainty of this point.

A prediction plot (cf. Fig. 3) shows the test cycles against the test target and the prediction target with and without uncertainty consideration. To summarize the performance of the used classification algorithm, a confusion matrix (cf. Fig. 4) is used. The classification error without considering uncertainty values is 0 % whereas the consideration of uncertainty leads to the conclusion that for 4.86 % (resp. 7 cycles) the prediction is correct, however very susceptible to random noise. This leads to the conclusion, that in a real-world example the 0 % test error is unrealistic and an error rate up to 4.86 % can be expected

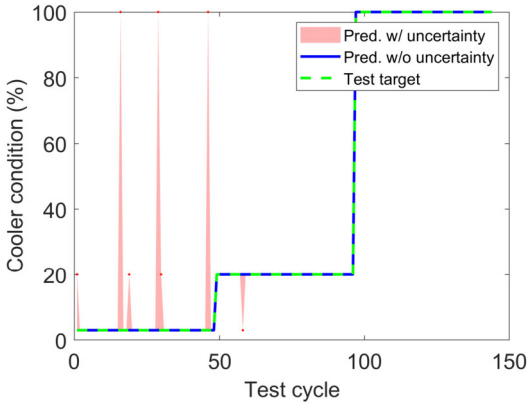


Figure 3: Prediction plot for test data with (red) and without (blue) consideration of uncertainty in contrast to the test target (green dashed).

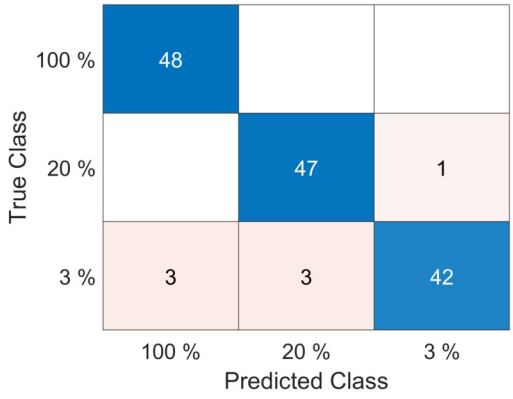


Figure 4: Confusion matrix for the cooler condition classification problem.

due to the shown susceptibility to noise. This also shows the benefits of uncertainty analysis in machine learning, as it provides a more realistic estimate of the expected performance in the field and at the same time highlights weaknesses like noise susceptibility that could be used as leverage points for further model improvement.

4 Conclusion and future work

In this work, the AMLT presented in [1] and [2] was extended inspired by some principles outlined in the GUM. Analytical approaches are presented for four of the five feature extraction methods either by literature references or in detail as for the statistical moments method. As the analytical approach leads to computational problems for the PCA, an efficient Monte Carlo implementation is used for the uncertainty calculation. In the feature selection step, filter methods expanded by weights are introduced and an extension of a standard SVM is used as wrapper method. For the classification step, the uncertainty propagation, especially for the LDA, is mathematically explained in detail. The code for this uncertainty-aware AMLT can be found on GitHub (https://github.com/ZeMA-gGmbH/LMT-UA-ML-Toolbox). Thereby, the determination of measurement uncertainty does not have to be regarded as an additional burden, but as a worthwhile addition with added value. For instance, with the extended AMLT, it was shown by taking measurement uncertainty for the sensor data into account, that there is an influence of measurement uncertainty on the model-based results. This influence will be investigated further in future work.

Funding: Part of this work has received funding within the project 17IND12 Met4FoF from the EMPIR program co-financed by the Participating States and from the European Union's Horizon 2020 research and innovation program. The basic version of the automated ML toolbox was developed at ZeMA as part of the MoSeS-Pro research project funded by the German Federal Ministry of Education and Research in the call "Sensor-based electronic systems for applications for Industry 4.0 – SElekt I 4.0", funding code 16ES0419K, within the framework of the German Hightech Strategy.

Appendix A. Derivations of the sensitivity coefficients and the covariance matrix for statistical moments

A.1 Standard deviation

$$
\begin{aligned}
\beta_{p,j} &= \frac{\partial \sigma_p}{\partial d_j} \\
&= \frac{1}{2} \cdot \left(\frac{1}{N_p - 1} \sum_{i=a_p}^{e_p} (d_i - \overline{d_p})^2 \right)^{-\frac{1}{2}} \\
&\quad \cdot \frac{2}{N_p - 1} \cdot \sum_{i=a_p}^{e_p} \left((d_i - \overline{d_p}) \cdot \frac{\partial}{\partial d_j} (d_i - \overline{d_p}) \right) \\
&= \frac{1}{2} \cdot \sigma_p^{-1} \cdot \frac{2}{N_p - 1} \\
&\quad \cdot \left((d_j - \overline{d_p}) \cdot \left(1 - \frac{1}{N_p} \right) + \sum_{i=a_p, i \neq j}^{e_p} (d_i - \overline{d_p}) \cdot \left(-\frac{1}{N_p} \right) \right)
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{1}{2} \cdot \sigma_p^{-1} \cdot \frac{2}{N_p - 1} \cdot \left((d_j - \overline{d_p}) - \frac{1}{N_p} \sum_{i=a_p}^{e_p} (d_i - \overline{d_p}) \right) \\
&= \frac{1}{2} \cdot \sigma_p^{-1} \cdot \frac{2}{N_p - 1} \cdot \left(d_j - \overline{d_p} - \frac{1}{N_p} \left(\sum_{i=a_p}^{e_p} d_i - N_p \overline{d_p} \right) \right) \\
&= \frac{1}{2} \cdot \sigma_p^{-1} \cdot \frac{2}{N_p - 1} \cdot \left(d_j - \overline{d_p} - \frac{1}{N_p} \sum_{i=a_p}^{e_p} d_i + \frac{N_p}{N_p} \overline{d_p} \right) \\
&= \frac{1}{2} \cdot \sigma_p^{-1} \cdot \frac{2}{N_p - 1} \cdot (d_j - \overline{d_p} - \overline{d_p} + \overline{d_p}) \\
&= \frac{1}{2 \cdot \sigma_p} \cdot \frac{2}{N_p - 1} \cdot (d_j - \overline{d_p}) \\
&= \frac{d_j - \overline{d_p}}{(N_p - 1) \cdot \sigma_p}
\end{aligned}
$$

A.2 Skewness

$$
\begin{aligned}
\frac{\partial v_p^{\text{denom}}}{\partial d_j} &= \frac{3}{N_p} \cdot \sum_{i=a_p}^{e_p} \left((d_i - \overline{d_p})^2 \cdot \frac{\partial}{\partial d_j} (d_i - \overline{d_p}) \right) \\
&= \frac{3}{N_p} \cdot \left((d_j - \overline{d_p})^2 \cdot \left(1 - \frac{1}{N_p} \right) \right. \\
&\quad \left. + \sum_{i=a_p, i \neq j}^{e_p} (d_i - \overline{d_p})^2 \cdot \left(-\frac{1}{N_p} \right) \right) \\
&= \frac{3}{N_p} \cdot \left((d_j - \overline{d_p})^2 - \frac{1}{N_p} \sum_{i=a_p}^{e_p} (d_i - \overline{d_p})^2 \right) \\
\frac{\partial v_p^{\text{nom}}}{\partial d_j} &= \frac{3}{2} \cdot \left(\frac{1}{N_p} \sum_{i=a_p}^{e_p} (d_i - \overline{d_p})^2 \right)^{\frac{1}{2}} \\
&\quad \cdot \frac{2}{N_p} \cdot \sum_{i=a_p}^{e_p} \left((d_i - \overline{d_p})^1 \cdot \frac{\partial}{\partial d_j} (d_i - \overline{d_p}) \right) \\
&= \frac{3}{2} \cdot \left(\frac{1}{N_p} \sum_{i=a_p}^{e_p} (d_i - \overline{d_p})^2 \right)^{\frac{1}{2}} \\
&\quad \cdot \frac{2}{N_p} \cdot \left((d_j - \overline{d_p}) \cdot \left(1 - \frac{1}{N_p} \right) \right. \\
&\quad \left. + \sum_{i=a_p, i \neq j}^{e_p} (d_i - \overline{d_p}) \cdot \left(-\frac{1}{N_p} \right) \right) \\
&= \frac{3}{2} \cdot \left(\frac{1}{N_p} \sum_{i=a_p}^{e_p} (d_i - \overline{d_p})^2 \right)^{\frac{1}{2}} \\
&\quad \cdot \frac{2}{N_p} \cdot \left((d_j - \overline{d_p}) - \frac{1}{N_p} \sum_{i=a_p}^{e_p} (d_i - \overline{d_p}) \right) \\
&= \frac{3}{2} \cdot \left(\frac{1}{N_p} \sum_{i=a_p}^{e_p} (d_i - \overline{d_p})^2 \right)^{\frac{1}{2}} \\
&\quad \cdot \frac{2}{N_p} \cdot \left(d_j - \overline{d_p} - \frac{1}{N_p} \sum_{i=a_p}^{e_p} d_i + \frac{N_p}{N_p} \cdot \overline{d_p} \right) \\
&= \frac{3}{N_p} \cdot \left(\frac{1}{N_p} \sum_{i=a_p}^{e_p} (d_i - \overline{d_p})^2 \right)^{\frac{1}{2}} \cdot (d_j - \overline{d_p})
\end{aligned}
$$

A.3 Kurtosis

$$
\begin{aligned}
\frac{\partial w_p^{\text{denom}}}{\partial d_j} &= \frac{4}{N_p} \cdot \sum_{i=a_p}^{e_p} \left((d_i - \overline{d_p})^3 \cdot \frac{\partial}{\partial d_j} (d_i - \overline{d_p}) \right) \\
&= \frac{4}{N_p} \cdot \left((d_j - \overline{d_p})^3 \cdot \left(1 - \frac{1}{N_p} \right) \right. \\
&\quad \left. + \sum_{i=a_p, i \neq j}^{e_p} (d_i - \overline{d_p})^3 \cdot \left(-\frac{1}{N_p} \right) \right) \\
&= \frac{4}{N_p} \cdot \left((d_j - \overline{d_p})^3 - \frac{1}{N_p} \sum_{i=a_p}^{e_p} (d_i - \overline{d_p})^3 \right) \\
\frac{\partial w_p^{\text{nom}}}{\partial d_j} &= 2 \cdot \left(\frac{1}{N_p} \sum_{i=a_p}^{e_p} (d_i - \overline{d_p})^2 \right)^1 \\
&\quad \cdot \frac{2}{N_p} \cdot \sum_{i=a_p}^{e_p} \left((d_i - \overline{d_p})^1 \cdot \frac{\partial}{\partial d_j} (d_i - \overline{d_p}) \right) \\
&= 2 \cdot \left(\frac{1}{N_p} \sum_{i=a_p}^{e_p} (d_i - \overline{d_p})^2 \right) \cdot \frac{2}{N_p} \\
&\quad \cdot \left((d_j - \overline{d_p}) \cdot \left(1 - \frac{1}{N_p} \right) \right. \\
&\quad \left. + \sum_{i=a_p, i \neq j}^{e_p} (d_i - \overline{d_p}) \cdot \left(-\frac{1}{N_p} \right) \right) \\
&= 2 \cdot \left(\frac{1}{N_p} \sum_{i=a_p}^{e_p} (d_i - \overline{d_p})^2 \right) \\
&\quad \cdot \frac{2}{N_p} \cdot \left((d_j - \overline{d_p}) - \frac{1}{N_p} \sum_{i=a_p}^{e_p} (d_i - \overline{d_p}) \right) \\
&= 2 \cdot \left(\frac{1}{N_p} \sum_{i=a_p}^{e_p} (d_i - \overline{d_p})^2 \right) \\
&\quad \cdot \frac{2}{N_p} \cdot \left(d_j - \overline{d_p} - \frac{1}{N_p} \sum_{i=a_p}^{e_p} d_i + \frac{N_p}{N_p} \cdot \overline{d_p} \right) \\
&= \frac{4}{N_p^2} \cdot \left(\sum_{i=a_p}^{e_p} (d_i - \overline{d_p})^2 \right) \cdot (d_j - \overline{d_p})
\end{aligned}
$$

A.4 Covariance matrix

$$
\begin{aligned}
\mathbf{U}^q &= \mathbf{J}^q_{\alpha,\beta,\gamma,\delta} \cdot \mathbf{U_c} \cdot \left(\mathbf{J}^q_{\alpha,\beta,\gamma,\delta}\right)^\top \\
&= \begin{pmatrix} \mathbf{A} \\ \mathbf{B} \\ \mathbf{\Gamma} \\ \mathbf{\Delta} \end{pmatrix} \cdot \mathbf{U_c} \cdot \begin{pmatrix} \mathbf{A}^\top, \mathbf{B}^\top, \mathbf{\Gamma}^\top, \mathbf{\Delta}^\top \end{pmatrix} \\
&= \begin{pmatrix} \mathbf{A} \\ \mathbf{B} \\ \mathbf{\Gamma} \\ \mathbf{\Delta} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{U_c}\mathbf{A}^\top, \mathbf{U_c}\mathbf{B}^\top, \mathbf{U_c}\mathbf{\Gamma}^\top, \mathbf{U_c}\mathbf{\Delta}^\top \end{pmatrix} \\
&= \begin{pmatrix} \mathbf{A}\mathbf{U_c}\mathbf{A}^\top & \mathbf{A}\mathbf{U_c}\mathbf{B}^\top & \mathbf{A}\mathbf{U_c}\mathbf{\Gamma}^\top & \mathbf{A}\mathbf{U_c}\mathbf{\Delta}^\top \\ \mathbf{B}\mathbf{U_c}\mathbf{A}^\top & \mathbf{B}\mathbf{U_c}\mathbf{B}^\top & \mathbf{B}\mathbf{U_c}\mathbf{\Gamma}^\top & \mathbf{B}\mathbf{U_c}\mathbf{\Delta}^\top \\ \mathbf{\Gamma}\mathbf{U_c}\mathbf{A}^\top & \mathbf{\Gamma}\mathbf{U_c}\mathbf{B}^\top & \mathbf{\Gamma}\mathbf{U_c}\mathbf{\Gamma}^\top & \mathbf{\Gamma}\mathbf{U_c}\mathbf{\Delta}^\top \\ \mathbf{\Delta}\mathbf{U_c}\mathbf{A}^\top & \mathbf{\Delta}\mathbf{U_c}\mathbf{B}^\top & \mathbf{\Delta}\mathbf{U_c}\mathbf{\Gamma}^\top & \mathbf{\Delta}\mathbf{U_c}\mathbf{\Delta}^\top \end{pmatrix} \\
&= \begin{pmatrix} \mathbf{A}\mathbf{U_c}\mathbf{A}^\top & \mathbf{A}\mathbf{U_c}\mathbf{B}^\top & \mathbf{A}\mathbf{U_c}\mathbf{\Gamma}^\top & \mathbf{A}\mathbf{U_c}\mathbf{\Delta}^\top \\ (\mathbf{A}\mathbf{U_c}\mathbf{B}^\top)^\top & \mathbf{B}\mathbf{U_c}\mathbf{B}^\top & \mathbf{B}\mathbf{U_c}\mathbf{\Gamma}^\top & \mathbf{B}\mathbf{U_c}\mathbf{\Delta}^\top \\ (\mathbf{A}\mathbf{U_c}\mathbf{\Gamma}^\top)^\top & (\mathbf{B}\mathbf{U_c}\mathbf{\Gamma}^\top)^\top & \mathbf{\Gamma}\mathbf{U_c}\mathbf{\Gamma}^\top & \mathbf{\Gamma}\mathbf{U_c}\mathbf{\Delta}^\top \\ (\mathbf{A}\mathbf{U_c}\mathbf{\Delta}^\top)^\top & (\mathbf{B}\mathbf{U_c}\mathbf{\Delta}^\top)^\top & (\mathbf{\Gamma}\mathbf{U_c}\mathbf{\Delta}^\top)^\top & \mathbf{\Delta}\mathbf{U_c}\mathbf{\Delta}^\top \end{pmatrix}
\end{aligned}
$$

References

1. Tizian Schneider, Nikolai Helwig, and Andreas Schütze. Industrial condition monitoring with smart sensors using automated feature extraction and selection. *Measurement Science and Technology*, 29(9), 2018.
2. Tanja Dorst, Yannick Robin, Tizian Schneider, and Andreas Schütze. Automated ML Toolbox for Cyclic Sensor Data. In *MSMM 2021 – Mathematical and Statistical Methods for Metrology*, pages 149–150, Online, Jun 2021.
3. Ronald Aylmer Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, Sep 1936.
4. Pourya Shamsolmoali, Deepak Kumar Jain, Masoumeh Zareapoor, Jie Yang, and M Afshar Alam. High-dimensional multimedia classification using deep CNN and extended residual units. *Multimedia Tools and Applications*, 78(17):23867–23882, 2019.
5. Douglas M Hawkins. The problem of overfitting. *Journal of Chemical Information and Computer Sciences*, 44(1):1–12, Jan 2004.
6. Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When Is "Nearest Neighbor" Meaningful? In *Database Theory — ICDT'99*, pages 217–235. Springer Berlin Heidelberg, 1999.
7. Michel Verleysen and Damien François. The Curse of Dimensionality in Data Mining and Time Series Prediction. In Joan Cabestany, Alberto Prieto, and Francisco Sandoval, editors, *Computational Intelligence and Bioinspired Systems*, pages 758–770. Springer Berlin Heidelberg, 2005.
8. Dimitrios Stratakis, Andreas Miaoudakis, Charalambos Katsidis, Vassilios Zacharopoulos, and Thomas Xenos. On the uncertainty estimation of electromagnetic field measurements using field sensors: a general approach. *Radiation Protection Dosimetry*, 133(4):240–247, 2009.
9. Maximilian Gruber, Wenzel Pilar von Pilchau, Varun Gowtham, Nikolaos-Stefanos Koutrakis, Matthias Riedl, Sascha Eichstädt, Jörg Hähner, Eckart Uhlmann, Julian Polte, and Alexander Willner. Uncertainty-Aware Sensor Fusion in Sensor Networks. In *SMSI 2021 – Sensor and Measurement Science International*, pages 346–347, 2021.
10. Robert T. Olszewski, Roy A. Maxion, and Dan P. Siewiorek. *Generalized feature extraction for structural pattern recognition in time-series data*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2001.
11. Ingrid Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992.
12. Fabian Mörchen. Time series feature extraction for data mining using DWT and DFT. *Department of Mathematics and Computer Science, University of Marburg, Germany – Technical Report*, 33:1–31, 2003.
13. Karl Pearson F. R. S.. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.
14. Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, 1933.
15. Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1-3):37–52, 1987. Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists.
16. J. Edward Jackson. *A Use's Guide to Principal Components*. John Wiley & Sons, Inc., 1991.
17. H. R. Martin and Farhang Honarvar. Application of statistical moments to bearing failure detection. *Applied Acoustics*, 44(1):67–77, 1995.
18. Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, Mar 2003.
19. Alain Rakotomamonjy. Variable selection using SVM-based criteria. *Journal of Machine Learning Research*, 3:1357–1370, Mar 2003.
20. Kenji Kira and Larry A. Rendell. The Feature Selection Problem: Traditional Methods and a New Algorithm. In *Proceedings / Tenth National Conference on Artificial Intelligence*, July 12–16, 1992, pages 129–134. AAAI Press, 1992.
21. Kenji Kira and Larry A. Rendell. A Practical Approach to Feature Selection. In Derek Sleeman and Peter Edwards, editors, *Machine Learning Proceedings 1992*, pages 249–256. Morgan Kaufmann, San Francisco (CA), 1992.
22. Igor Kononenko and Se June Hong. Attribute selection for modelling. *Future Generation Computer Systems*, 13(2-3):181–195, Nov 1997.
23. Igor Kononenko, Edvard Šimec, and Marko Robnik-Šikonja. Overcoming the myopia of inductive learning algorithms with RELIEFF. *Applied Intelligence*, 7(1):39–55, Jan 1997.
24. Marko Robnik-Šikonja and Igor Kononenko. Theoretical and empirical analysis of ReliefF and RReliefF. *Machine Learning*, 53(1):23–69, 2003.
25. Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern*

Classification, 2 edition. A Wiley-Interscience Publication. Wiley, New York, 2001.
26. Prasanta Chandra Mahalanobis. On tests and measures of group divergence. *Journal of the Asiatic Society of Bengal*, 26:541–588, 1930.
27. Prasanta Chandra Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, 2:49–55, 1936.
28. Roy De Maesschalck, Delphine Jouan-Rimbaud, and Desire L. Massart. The Mahalanobis distance. *Chemometrics and Intelligent Laboratory Systems*, 50(1):1–18, 2000.
29. Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence – Volume 2, IJCAI '95*, pages 1137–1143. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995.
30. BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP, and OIML. JCGM 100: Evaluation of measurement data Guide to the expression of uncertainty in measurement. 2008.
31. BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP, and OIML. JCGM 101: Evaluation of measurement data Supplement 1 to the "Guide to the expression of uncertainty in measurement" Propagation of distributions using a Monte Carlo method. 2008.
32. BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP, and OIML. JCGM 102: Evaluation of measurement data Supplement 2 to the "Guide to the expression of uncertainty in measurement" Extension to any number of output quantities. 2011.
33. Sascha Eichstädt, Alfred Link, Peter Harris, and Clemens Elster. Efficient implementation of a Monte Carlo method for uncertainty evaluation in dynamic measurements. *Metrologia*, 49(3):401–410, Apr 2012.
34. Tanja Dorst, Sascha Eichstädt, Tizian Schneider, and Andreas Schütze. Propagation of uncertainty for an Adaptive Linear Approximation algorithm. In *SMSI 2020 – Sensor and Measurement Science International*, pages 366–367. Jun 2020.
35. Tanja Dorst, Sascha Eichstädt, Tizian Schneider, and Andreas Schütze. GUM2ALA – Uncertainty propagation algorithm for the Adaptive Linear Approximation according to the GUM. In *SMSI 2021 – Sensor and Measurement Science International*, pages 314–315, May 2021.
36. Sascha Eichstädt and Volker Wilkens. GUM2DFT — a software tool for uncertainty evaluation of transient signals in the frequency domain. *Measurement Science and Technology*, 27(5):055001, 2016.
37. Lorenzo Peretto, Renato Sasdelli, and Roberto Tinarelli. Uncertainty propagation in the discrete-time wavelet transform. In *Proceedings of the 20th IEEE Instrumentation Technology Conference (Cat. No. 03CH37412), volume 2*, pages 1465–1470, 2003.
38. Lorenzo Peretto, Renato Sasdelli, and Roberto Tinarelli. Uncertainty propagation in the discrete-time wavelet transform. *IEEE Transactions on Instrumentation and Measurement*, 54(6):2474–2480, 2005.
39. Lorenzo Peretto, Renato Sasdelli, and Roberto Tinarelli. On uncertainty in wavelet-based signal analysis. *IEEE Transactions on Instrumentation and Measurement*, 54(4):1593–1599, 2005.
40. Maximilian Gruber, Tanja Dorst, Andreas Schütze, Sascha Eichstädt, and Clemens Elster. Discrete wavelet transform on uncertain data: Efficient online implementation for practical applications. In Franco Pavese, Alistair B Forbes, Nien-Fan Zhang, and Anna Chunovkina, editors, *Series on Advances in Mathematics for Applied Sciences*, pages 249–261. World Scientific, Jan 2022.
41. Yingyao Zhou, Jason A. Young, Andrey Santrosyan, Kaisheng Chen, Frank S. Yan, and Elizabeth A. Winzeler. In silico gene function prediction using ontology-based pattern identification. *Bioinformatics*, 21(7):1237–1245, Apr 2005.
42. Charles Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15:72–101, 1904.
43. Clark Wissler. The Spearman correlation formula. *Science*, 22(558):309–311, 1905.
44. Jinbo Bi and Tong Zhang. Support Vector Classification with Input Data Uncertainty. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2004.
45. Gene H. Golub and Charles F. van Loan. An analysis of the total least squares problem. *SIAM Journal on Numerical Analysis*, 17(6):883–893, 1980.
46. Roger A. Horn. The Hadamard product. In Charles R. Johnson, editor, *Matrix theory and applications*, volume 40 of *Proceedings of Symposia in Applied Mathematics*, pages 87–169. Amer. Math. Soc., Providence, RI, 1990.
47. Robert Reams. Hadamard inverses, square roots and products of almost semidefinite matrices. *Linear Algebra and its Applications*, 288:35–43, 1999.
48. Tizian Schneider, Steffen Klein, and Manuel Bastuck. Condition monitoring of hydraulic systems Data Set at ZeMA, Apr 2018.
49. Nikolai Helwig, Eliseo Pignanelli, and Andreas Schütze. Condition monitoring of a complex hydraulic system using multivariate statistics. In *2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, pages 210–215, 2015.
50. Nikolai Helwig, Eliseo Pignanelli, and Andreas Schütze. Detecting and Compensating Sensor Faults in a Hydraulic Condition Monitoring System. In *Proceedings SENSOR 2015*, pages 641–646, 2015.

Bionotes

Tanja Dorst
ZeMA – Center for Mechatronics and Automation Technology gGmbH, Saarbrücken, Germany
t.dorst@zema.de

Tanja Dorst studied Mathematics at Saarland University and received her Master of Science degree in November 2013. After that, she studied Mechanical Engineering at University of Applied Sciences in Saarbrücken and received her Bachelor of Engineering degree in September 2017. Since July 2020 she has been working at Center for Mechatronics and Automation Technology (ZeMA) gGmbH as a scientific researcher. Her research interests include measurement uncertainties in ML for condition monitoring of technical systems.

Tizian Schneider
ZeMA – Center for Mechatronics and Automation Technology gGmbH, Saarbrücken, Germany
Lab for Measurement Technology, Department of Mechatronics, Saarland University, Saarbrücken, Germany
t.schneider@zema.de

Tizian Schneider studied Microtechnologies and Nanostructures at Saarland University and received his Master of Science degree in January 2016. Since that time, he has been working at the Lab for Measurement Technology (LMT) of Saarland University and at Center for Mechatronics and Automation Technology (ZeMA) gGmbH leading the research group Data Engineering & Smart Sensors. His research interests include ML methods for condition monitoring of technical systems, automatic ML model building and interpretable AI.

Sascha Eichstädt
Physikalisch-Technische Bundesanstalt, Braunschweig and Berlin, Germany
sascha.eichstaedt@ptb.de

Dr. Sascha Eichstädt is the leader of the Physikalisch-Technische Bundesanstalt (PTB) department “Metrology for digital transformation”. He received his Diploma in Mathematics in 2008 at the HU Berlin, and his PhD in Theoretical Physics in 2012 at the TU Berlin. From 2008 to 2017 he joined the group “Mathematical modelling and data analysis” at PTB. His main research areas are signal processing and sensor networks.

Andreas Schütze
ZeMA – Center for Mechatronics and Automation Technology gGmbH, Saarbrücken, Germany
Lab for Measurement Technology, Department of Mechatronics, Saarland University, Saarbrücken, Germany
schuetze@lmt.uni-saarland.de

Andreas Schütze received his diploma in physics from RWTH Aachen in 1990 and his doctorate in Applied Physics from Justus-Liebig-Universität in Gießen in 1994 with a thesis on microsensors and sensor systems for the detection of reducing and oxidizing gases. From 1994 until 1998 he worked for VDI/VDE-IT, Teltow, Germany, mainly in the fields of microsystems technology. From 1998 until 2000 he was professor for Sensors and Microsystem Technology at the University of Applied Sciences in Krefeld, Germany. Since April 2000 he is professor for Measurement Technology in the Department Systems Engineering at Saarland University, Saarbrücken, Germany and head of the Laboratory for Measurement Technology (LMT). His research interests include smart gas sensor systems as well as data engineering methods for industrial applications.

6.3 Paper D: Influence of Synchronization within a Sensor Network on Machine Learning Results

Another potential issue for ML-based condition monitoring investigated in the following paper is synchronization errors in sensor networks. As data is typically recorded by multiple sensors and potentially by different data acquisition systems, their exact synchronization is essential and lack of such can cause issues. To investigate those influences, a trigger-synchronized dataset for remaining useful lifetime estimation is altered with artificially added random time shifts between sensors to quantify the performance degradation that became significant for random shifts larger than 0.1 ms. Constant time shifts did not alter the performance.

Additionally, the features the resulting prediction model is based on were physically interpreted, and two strategies for time-shift compensations were compared. Those strategies are modifying feature extraction to use time shift-invariant features and training with augmented data that included time shifts.

Influence of Synchronization within a Sensor Network on Machine Learning Results

Tanja Dorst[1], Yannick Robin[2], Sascha Eichstädt[3], Andreas Schütze[2], and Tizian Schneider[1]

[1]*Centre for Mechatronics and Automation Technology (ZeMA gGmbH), Saarbrücken, Germany*

[2]*Saarland University, Lab for Measurement Technology, Saarbrücken, Germany*

[3]*Physikalisch-Technische Bundesanstalt PTB, Braunschweig, Germany*

Journal of Sensors and Sensor Systems (2021), 10 (2), 233-245

The original paper can be found online at https://doi.org/10.5194/jsss-10-233-2021.

© 2021 by the authors. This open-access article is distributed under the terms and conditions of the Creative Commons Attribution 4.0 License. (http://creativecommons.org/licenses/by/4.0/).

J. Sens. Sens. Syst., 10, 233–245, 2021
https://doi.org/10.5194/jsss-10-233-2021
© Author(s) 2021. This work is distributed under
the Creative Commons Attribution 4.0 License.

JSSS Journal of Sensors and Sensor Systems Open Access

Influence of synchronization within a sensor network on machine learning results

Tanja Dorst[1], Yannick Robin[2], Sascha Eichstädt[3], Andreas Schütze[1,2], and Tizian Schneider[1,2]

[1]ZeMA – Center for Mechatronics and Automation Technology gGmbH, Saarbrücken, Germany
[2]Lab for Measurement Technology, Department of Mechatronics, Saarland University, Saarbrücken, Germany
[3]Physikalisch-Technische Bundesanstalt, Braunschweig and Berlin, Germany

Correspondence: Tanja Dorst (t.dorst@zema.de)

Received: 10 March 2021 – Revised: 28 July 2021 – Accepted: 30 July 2021 – Published: 24 August 2021

Abstract. Process sensor data allow for not only the control of industrial processes but also an assessment of plant conditions to detect fault conditions and wear by using sensor fusion and machine learning (ML). A fundamental problem is the data quality, which is limited, inter alia, by time synchronization problems. To examine the influence of time synchronization within a distributed sensor system on the prediction performance, a test bed for end-of-line tests, lifetime prediction, and condition monitoring of electromechanical cylinders is considered. The test bed drives the cylinder in a periodic cycle at maximum load, a 1 s period at constant drive speed is used to predict the remaining useful lifetime (RUL). The various sensors for vibration, force, etc. integrated into the test bed are sampled at rates between 10 kHz and 1 MHz. The sensor data are used to train a classification ML model to predict the RUL with a resolution of 1 % based on feature extraction, feature selection, and linear discriminant analysis (LDA) projection. In this contribution, artificial time shifts of up to 50 ms between individual sensors' cycles are introduced, and their influence on the performance of the RUL prediction is investigated. While the ML model achieves good results if no time shifts are introduced, we observed that applying the model trained with unmodified data only to data sets with time shifts results in very poor performance of the RUL prediction even for small time shifts of 0.1 ms. To achieve an acceptable performance also for time-shifted data and thus achieve a more robust model for application, different approaches were investigated. One approach is based on a modified feature extraction approach excluding the phase values after Fourier transformation; a second is based on extending the training data set by including artificially time-shifted data. This latter approach is thus similar to data augmentation used to improve training of neural networks.

1 Introduction

In the Industry 4.0 paradigm, industrial companies have to deal with several emerging challenges of which digitalization of the factory is one of the most important aspects for success. In digitalized factories, sometimes also referred to as "Factories of the Future" (FoF), the "Industrial Internet of Things" (IIoT) forms the networking basis and allows users to improve operational effectiveness and strategic flexibility (Eichstädt, 2020; Schütze et al., 2018). Key components of FoF and IIoT are intelligent sensor systems, also called cyber-physical systems, and machine learning (ML), which allow for the automation and improvement of complex process and business decisions in a wide range of application areas. For example, smart sensors can be used to evaluate the state of various components, determine the optimum maintenance schedule, or detect fault conditions (Schneider et al., 2018b), as well as to control entire production lines (Usuga Cadavid et al., 2020). To make full use of the wide-ranging potential of smart sensors, the quality of sensor data has to be taken into account (Teh et al., 2020). This is limited by environmental factors, sensor failures, measurement uncertainty, and – especially in distributed sensor networks – by time synchronization errors between individual sensors. Confidence in ML algorithms and their decisions or predictions requires reliable data and therefore a metrological in-

frastructure allowing for an assessment of the data quality. In this contribution, a software toolbox for statistical machine learning (Schneider et al., 2017, 2018b; Dorst et al., 2021a) is used to evaluate large data sets from distributed sensor networks under the influence of artificially generated time shifts to simulate synchronization errors. One aspect to address time synchronization problems in distributed sensor networks is improved time synchronization methods to provide a reliable global time for all sensors. Many different synchronization methods are proposed for sensor networks (Sivrikaya and Yener, 2004). However, improved time synchronization might not be possible or be too costly, especially in existing sensor networks which were often never designed for sensor data fusion, so the ML approach can be improved to achieve a more robust model with acceptable results as demonstrated in this contribution.

2 Test bed for data acquisition

Predictive maintenance, based on reliable condition monitoring, is a requirement for reducing repair costs and machine downtime and, as a consequence, increasing productivity. Therefore, an estimation of the remaining useful lifetime (RUL) of critical components is required. Since we are using a data-driven model, this cannot be done directly without reference data. A test bed for electromechanical cylinders (EMCs) with a spindle drive equipped with several sensors is used. This specific test bed was used as it contains a large variety of sensor domains and allows for physical interpretation. Because most industrial ML problems only use a subset of these sensors, the approaches of the chosen test bed can be transferred. In this test bed, long-term speed driving and high load tests are carried out until a position error of the EMC occurs, i.e., until the device under test (DUT) fails. Characteristic signal patterns and relevant sensors can be identified for condition monitoring as well as for RUL estimation of the EMCs. Figure 1 shows the scheme of the test bed. Simplified, the setup of the test bed consists of the tested EMC and a pneumatic cylinder which simulates the variable load on the EMC in axial direction. All parameters of the working cycle can be set by using a LabVIEW GUI.

A typical working cycle lasts 2.8 s. It consists of a forward stroke and a return stroke of the EMC as well as a waiting time of 150 ms between both linear movements. The movements are always carried out with approximately maximum speed and maximum acceleration. The stroke range of the EMC is between 100 and 350 mm in the test bed. The combination of high travel speed (200 mm s^{-1}), high axial force (7 kN), and high acceleration (5 mm s^{-2}) leads to fast wear of the EMC. The error criterion for failure of the EMC is defined as a too large deviation between the nominal and actual position values; i.e., the test is stopped as soon as the specified position accuracy (position accuracy < 30 mm) is no longer fulfilled due to increased friction.

To gather as much data as possible from different sensor domains for a comprehensive condition monitoring, the following 11 sensors are used within the test bed (Schneider et al., 2018a):

- one microphone with a sampling rate of 100 kHz;
- three accelerometers with 100 kHz sampling rate, attached at the plain bearing, at the piston rod, and at the ball bearing;
- four process sensors (axial force, pneumatic pressure, velocity, and active current of the EMC motor) with 10 kHz sampling rate each;
- three electrical motor current sensors with 1 MHz sampling rate each.

In Fig. 2, the raw data for one cycle and all sensors is shown. The collected data reflect the functionality of the EMC and its decrease during the long-term test. For data analysis, which is described in more detail in the next section, various EMCs were tested until the position error occurred. The typical lifetime of an EMC under these test conditions was approx. 629 000 cycles corresponding to roughly 20 d and generated an average of 12 TB of raw data.

3 ML toolbox for data analysis

The ML toolbox developed by Schneider et al. (2018b) is used for RUL analysis in this contribution. It can be applied in a fully automated way, i.e., without expert knowledge and without a detailed physical model of the process. After acquisition of the raw data, feature extraction and selection as well as classification and evaluation are performed, as shown in Fig. 3.

3.1 Feature extraction

In the beginning, unsupervised feature extraction (FE) is performed, i.e., without knowledge of the group to which the individual work cycle belongs, in this case the current state of aging (RUL). Features are generated from the repeating working cycles of the raw data. As there is no method that works well for all applications, features are extracted from different domains by five complementary methods:

- *Adaptive linear approximation* (ALA) divides the cycles into approximately linear segments. For each linear segment, mean value and slope are extracted as features from the time domain (Olszewski et al., 2001).
- Using *principal component analysis* (PCA), projections on the principal components are determined and used as features, representing the overall signal (Wold et al., 1987).

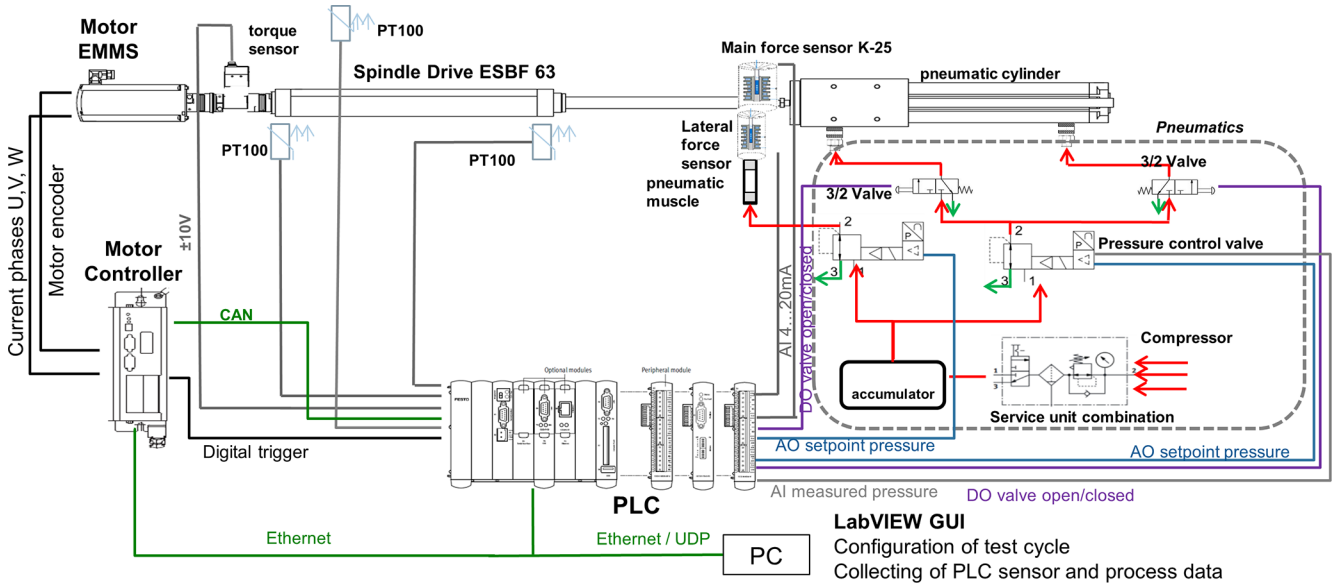


Figure 1. Basic scheme of the EMC test bed (Helwig et al., 2017).

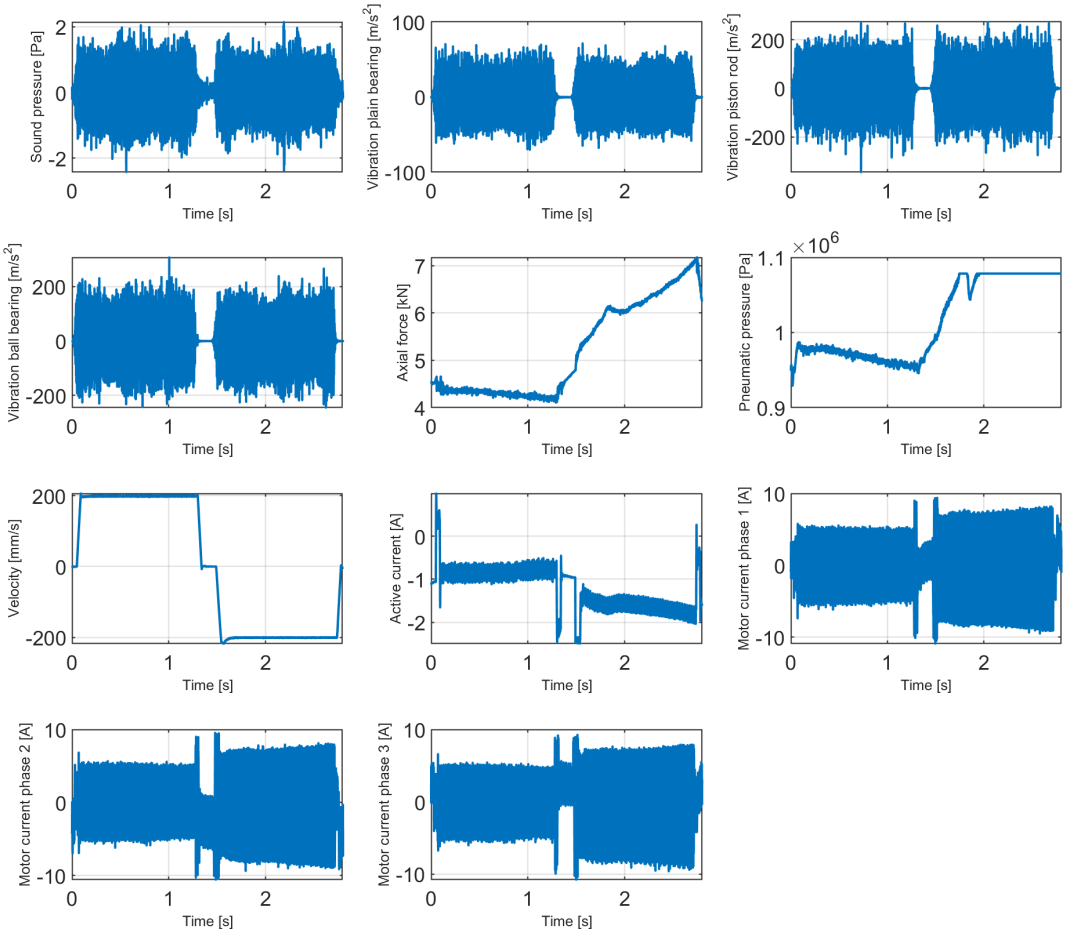


Figure 2. Raw data recorded during one cycle by 11 sensors expressed in SI units.

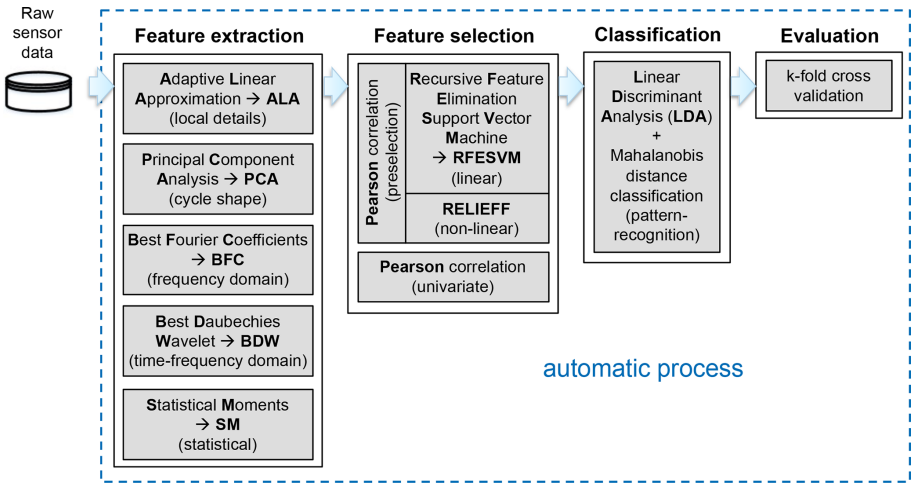


Figure 3. Schematic of the automatic toolbox for condition monitoring using machine learning, adapted from Dorst et al. (2019).

- The *best Fourier coefficient* (BFC) method extracts the 10 % of amplitudes with the highest average absolute value over all cycles and their corresponding phases as features from the frequency domain (Mörchen, 2003).
- The *best Daubechies wavelet* (BDW) algorithm is based on a wavelet transform, and as for BFC, the 10 % of the wavelet coefficients with the highest average absolute value over all cycles are chosen as features from the time-frequency domain.
- In general, information is also included in the statistical distribution of the measurement values. These features are extracted from a fixed number of equally sized segments of a cycle by the four *statistical moments* (SMs) of mean, variance, skewness, and kurtosis.

The objective of FE is to concentrate information in as few features as possible whilst achieving a precise prediction of the RUL. The FE methods are applied to all sensor signals and all cycles. This results in five feature sets with a large number of features in each. However, the number of features is still too high after performing feature extraction for Big Data applications, such as RUL estimation of the EMC as described in the previous section. Due to the insufficient data reduction in this step, feature selection is carried out with the extracted features to prevent the "curse of dimensionality" (Beyer et al., 1999).

3.2 Feature selection

Feature selection (FS) is a supervised step; i.e., the group to which each cycle belongs is known. In the case of the RUL estimation of the EMC, the target value is the used lifetime with a resolution of 1 %. As for feature extraction, no method alone can provide the optimum solution for all applications, so three different complementary methods are used for feature selection in the ML toolbox:

- *Recursive Feature Elimination Support Vector Machine* (RFESVM) uses a linear support vector machine (SVM) to recursively remove the features with the smallest contribution to the group separation from the set of all features (Guyon and Elisseeff, 2003; Rakotomamonjy, 2003).
- The *RELIEFF* algorithm is used when the groups cannot be separated linearly. This algorithm finds the nearest hits and nearest misses for each point by using k-nearest neighbors with the Manhattan norm (Kononenko and Hong, 1997; Robnik-Šikonja and Kononenko, 2003).
- *Pearson correlation* is used as a third method for feature (pre)selection because of its low computational cost. The features are sorted by their correlation coefficient to the target value. This coefficient indicates how large the linear correlation between a feature and the target value is.

Preselection based on Pearson correlation is performed to reduce the feature set to only 500 features before applying the RFESVM or RELIEFF algorithms to reduce the computational costs. After ranking the features with a feature selection algorithm, a 10-fold cross-validation (explained later) is carried out for every number of features to find the optimum number of features. Thus, the most relevant features with respect to the classification task are selected, and features with redundant or no information content are removed from the feature set.

In addition to reducing the data set, this step also avoids overfitting, which often occurs when the number of data points for developing the classification model is not significantly greater than the number of features.

3.3 Classification

The classification is carried out in two steps: a further dimensionality reduction followed by the classification itself. The further dimensionality reduction is based on *linear discriminant analysis* (LDA). It performs a linear projection of the feature space into a $g-1$-dimensional subspace for g groups which represent the corresponding system state. The intraclass variance, the variance within the classes, is minimized while the interclass variance, the variance between the classes, is maximized (Duda et al., 2001). Thus, the distance calculation in the classification step has only a complexity of $g-1$. The actual classification is carried out using the Mahalanobis distance; see Eq. (1):

$$d_{\mathrm{Mahal}}(\boldsymbol{x}) = \sqrt{(\boldsymbol{x}-\boldsymbol{m})^{\top}\mathbf{S}^{-1}(\boldsymbol{x}-\boldsymbol{m})}. \quad (1)$$

Here $\boldsymbol{x}$ denotes the vector of the test data, $\boldsymbol{m}$ the componentwise arithmetic mean, and $\mathbf{S}$ the covariance matrix of the group. For each data point, the Mahalanobis distance indicates how far it is away from the center of the data group, taking the group scattering into account. In order to classify the data, each sample is labeled with the class that has the smallest Mahalanobis distance. Points of equal Mahalanobis distance from a center graphically form a hyperellipse in the $g-1$-dimensional LDA space.

3.4 Evaluation

The k-fold stratified cross-validation (CV) is used for evaluation (Kohavi, 1995). This means the data set is randomly divided into k subsets, with $k \in \mathbb{N}$. Stratified means that each of the k subsets has approximately the same class distribution as the whole feature set. In the ML toolbox, k is usually set to 10. Thus, one group forms the test data set and nine groups form the training data set, from which the ML model is generated.

3.5 Automated ML toolbox

The automatic ML toolbox compares the 15 combinations that are achieved by combining all feature extraction methods and all selection methods. The cross-validation error, i.e., the percentage of misclassified cycles by the 10-fold cross-validation, is automatically calculated for each of the 10 permutations resulting from the 10-fold cross-validation and for each of the 15 FE/FS combinations. To compare the result of the different combinations, the mean of the 10 cross-validation errors (one cross-validation error per fold) per combination is used. The minimum value of all the 15 cross-validation errors (one error per combination) leads to the best combination of FE/FS method. Thus, finding the best combination of one feature extraction and one feature selection method for the current application case is a fully automated process that is performed offline. The actual classification is then carried out online by using only the best of the 15 combinations, which results in a low computational effort during application.

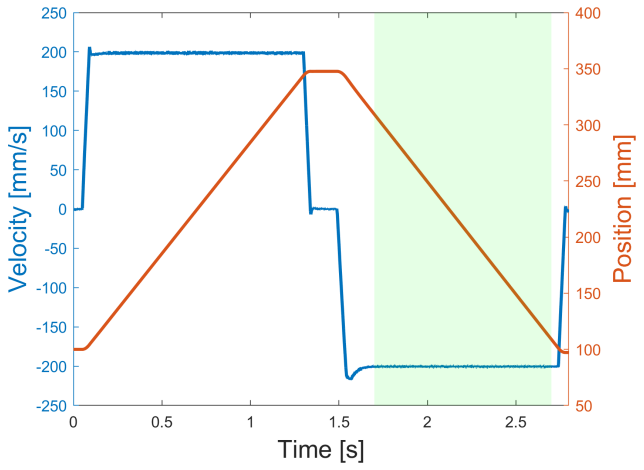


Figure 4. Working cycle depicted as position (red) and velocity (blue) consisting of forward stroke, waiting time, and return stroke, as well as the period (green) evaluated for estimation of the RUL.

4 Application of the ML toolbox on test bed data

The basis for this contribution is a lifetime test of an EMC which originally lasted 20.4 d and consists of 629 485 cycles. Only 1 s of the synchronous phase of the return stroke (duration 1.2 s) for each working cycle is evaluated with the ML toolbox. During this 1 s period, the velocity is constant and the load is highest as the EMC is pulling against a constant load provided by the pneumatic cylinder; see Fig. 4. Thus, this 1 s period is suitable for ML problems.

For this full data set, where all sensors have their original sampling rate, the minimum cross-validation error of 8.9 % was achieved with 499 features and a combination of BFC and Pearson correlation together with the previously described LDA classifier (Schneider et al., 2018c). Pearson correlation was only used as selector due to the high computational time of RFESVM and RELIEFF for the full data set with 629 485 cycles. Feature extraction together with feature selection leads to a data reduction of approximately a factor of 60 000 in this case; i.e., the originally recorded 12 TB of raw data for this EMC is reduced to a feature set of approximately 200 MB.

To reduce computational costs and to allow us to study various influencing factors on the classification performance, a reduced data set with only every hundredth cycle is used in this contribution. A further reduction of the computational costs could be achieved by reducing the sampling rate of the data. To test the influence of lower sampling rates, several data sets with different sampling rates are used, and it can be observed that the best results across all used sampling rates are always achieved with a combination of BFC and

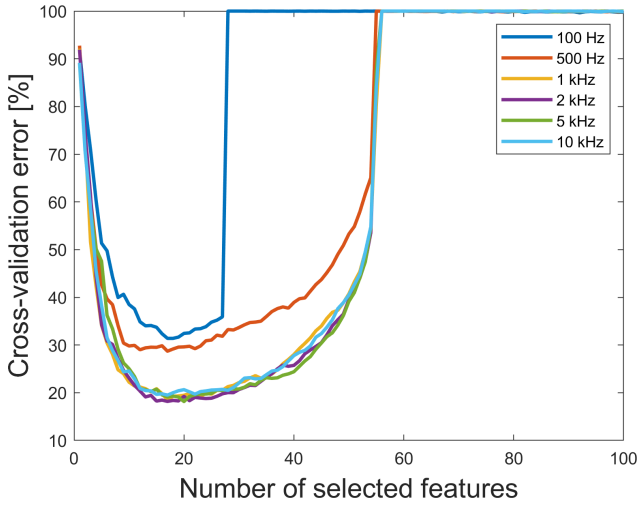


Figure 5. The 10-fold cross-validation error vs. number of selected features for data sets with different sampling rate using BFC as extractor and RFESVM as selector.

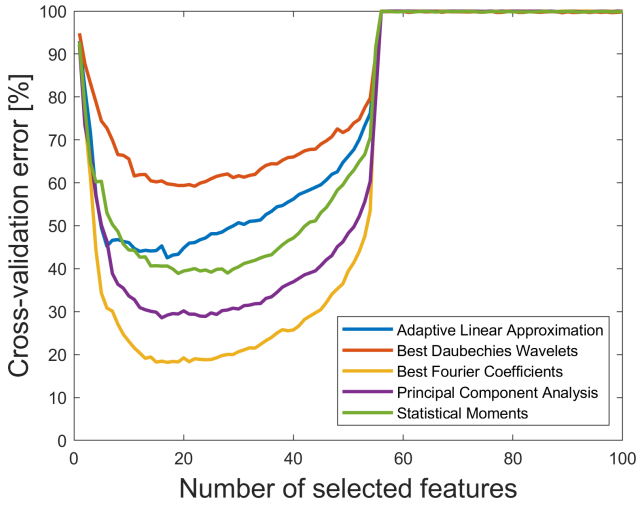


Figure 6. The 10-fold cross-validation error vs. number of selected features for the original 2 kHz data set without time shift using RFESVM as selector. For a better visibility, only results with RFESVM as selector are shown.

Table 1. Cross-validation error for different FE/FS combinations.

FS/FE	Pearson	RFESVM	RELIEFF
ALA	77.84 %	42.53 %	94.06 %
BDW	77.29 %	59.20 %	89.89 %
BFC	36.97 %	18.18 %	90.41 %
PCA	31.06 %	28.56 %	96.82 %
SM	57.91 %	38.89 %	99.05 %

RFESVM. As shown in Fig. 5, the minimum 10-fold cross-validation error of the EMC data sets with sampling rates of 1 kHz and more is nearly the same. Thus, the quality of the prediction is not influenced by a lower sampling rate. The minimum cross-validation error (18.15 %) is achieved with the 5 kHz data set, but with the 2 kHz version, the cross-validation error increases only slightly in the second decimal place (18.18 %). Thus, it is not necessary to use a data set with a higher sampling rate, and due to less computational costs, the 2 kHz data set is chosen for this contribution. It seems that several relevant features are in the range between 250 Hz and 1 kHz and, based on the Nyquist criterion, are thus contained in this data set. All further results in this contribution are based on the 2 kHz resolution data set of an EMC with 6292 cycles (1.1 GB) and time-shifted versions of this data set. The 2 kHz raw data set is available online for further analysis (Dorst, 2019).

For this data set, the lowest cross-validation error is reached with features extracted from the frequency domain with BFC and RFESVM as selector. The cross-validation error for the 15 FE/FS combinations can be found in Table 1.

The lowest cross-validation error with 18.18 % misclassifications occurs when using only 17 features as shown in Fig. 6. The large increase of the cross-validation error when using 54–56 features or more in Figs. 5 and 6 can be understood considering the covariance matrices $\mathbf{S}$ used for calculation of the Mahalanobis distance. These covariance matrices have a reciprocal condition number of about 10^{-19} in 1-norm, which means that they are ill-conditioned. A reason for the ill-conditioned covariance matrices is the low number of cycles (only 62, which results from the 1 % resolution of the RUL together with 6292 cycles) per target class and the nearly equal number of features.

Since 11 sensors are used within the test bed, Fig. 7 shows which sensors are contributing to the 17 most important features for the RUL prediction using BFC as the feature extractor and RFESVM as selector. It can be clearly seen that five features each (i.e., 29 %) are derived from the microphone and the active current data. For further analysis, it is important to note that 12 of the 17 best Fourier coefficient features represent amplitudes.

To check the plausibility of the results, Fig. 8 shows that these 17 most relevant features are within the range 0 to 640 Hz. Thus, using the 1 kHz data set would lead to a loss of relevant features (640 Hz). The dominant frequency here is 120 Hz (five features) which represents the third harmonic of the rotation frequency. The explanation for the other frequencies can be found in Table 2 (cf. Helwig, 2018).

5 Synchronization problems and their effects on machine learning results

Synchronization between different sensors is important to enable data analysis. Correctly performed data fusion is crucial for applications, e.g., in industrial condition monitoring (Helwig, 2018). Synchronization problems there simply means that the raw data of the sensors' cycles are shifted

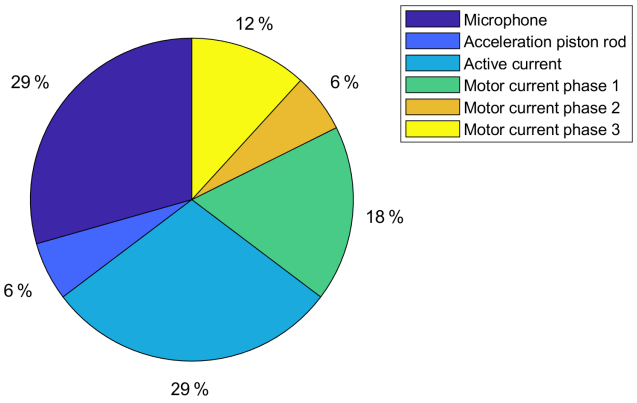


Figure 7. The 17 most important features by sensors, selected with RFESVM. Only 6 of the 11 sensors contribute to the 17 most important features.

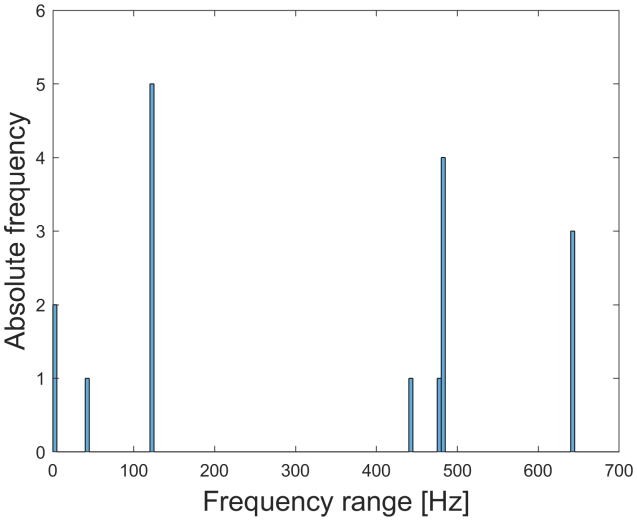


Figure 8. Frequency range of the 17 most relevant features. The frequencies of all relevant features are ≤ 640 Hz.

against each other. The feature extraction is carried out for every sensor and all features are packed together in the classifier. As the temporal localization of effects can play a role in ML, synchronization problems can lead to poor classification results like later shown in this contribution.

To analyze the effects of synchronization problems between the individual sensors installed within the test bed and their effect on the lifetime prognosis, time-shifted data sets downsampled to 2 kHz are used. Thereby, the raw data set with full resolution, mentioned in Sect. 4, serves as basis to simulate synchronization errors. These errors are simulated by manipulating the raw data set with random time shifts between the individual sensors' cycles in the 1.0 s window of the return stroke. The maximum time shift of a cycle is ± 50 ms in relation to the original time axis to ensure that only data from the return stroke are used for all sensors. The

Table 2. Explanation of the frequencies of the 17 most relevant features. The 17 most relevant features are physically explainable.

Frequency	Explanation
0 Hz	mean value of the signal
40 Hz	mechanical driving frequency
120 Hz	third harmonic of the rotation frequency
440 Hz	rollover frequency of the ball screw drive
480 Hz	damage frequency of the spindle nut
640 Hz	mechanical resonance

minimal possible time shift is ± 0.1 ms as the lowest sampling rate over all sensors is 10 kHz.

Clock synchronization is a topic of research still today (Yiğitler et al., 2020). As shown in this contribution, it is important to think about clock synchronization, because if not, then there will be serious issues with the results. For distributed sensor networks, the considered time shifts are in a range that can be expected (Tirado-Andrés and Araujo, 2019).

After simulating these errors with the raw data set, the different time-shifted data sets are downsampled to 2 kHz to reduce computational complexity. Analysis is carried out using time-shifted data sets with a minimum of ± 0.1 ms per cycle (based on the time axis of the 2 kHz raw data set) and sensor up to a maximum of ± 50 ms per cycle and sensor. The time-shifted values in every cycle for every sensor are randomly generated with a discrete uniform distribution. This means that the time shift for all samples of one single cycle is the same but not for the same cycle over all sensors. The best combination of FE/FS algorithm for all five time-shifted data sets is BFC as extractor together with RFESVM as selector. An increase in the cross-validation error is observed with increasing random time shifts for all sensors (cf. Table 3). For random time shifts between 0.1 and 1 ms, the cross-validation error is nearly the same; the change is only in the first decimal place. Using random time shifts with more than ± 50 ms leads to a significant decrease of the classification performance. A likely reason for this decrease is probably that not only data from the synchronous phase of the return stroke are used, but also some data from the acceleration or deceleration phase of the return stroke are included in the evaluated 1 s period. To depict the effect of increasing random time shifts on the prediction performance more clearly, the cross-validation error using BFC as extractor, RFESVM as selector, and time shifts from 0.1 to 50 ms between all 11 sensors are shown in Fig. 9 vs. the number of features. Every model was trained with the specific time-shifted data set. It can be clearly seen that small time shifts only have a minor effect on the cross-validation error, whereas time shifts of 1 ms or more increase the cross-validation error noticeably. One reason is that the variance in the data increases by increasing random time shifts and makes it harder for the

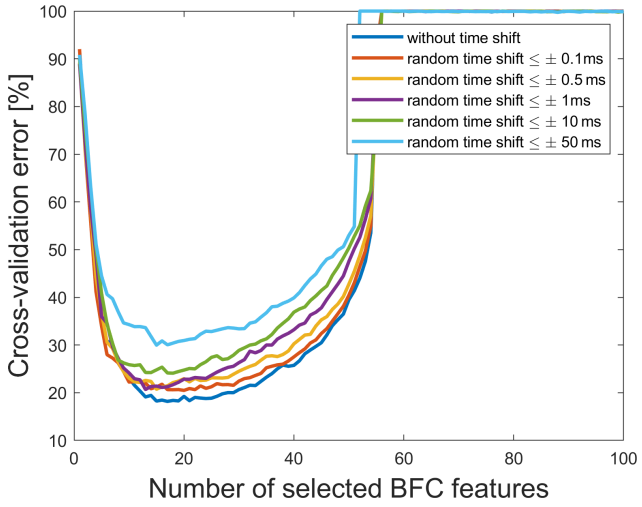


Figure 9. Cross-validation errors vs. the number of selected BFC features for different random simulated synchronization errors using RFESVM as selector.

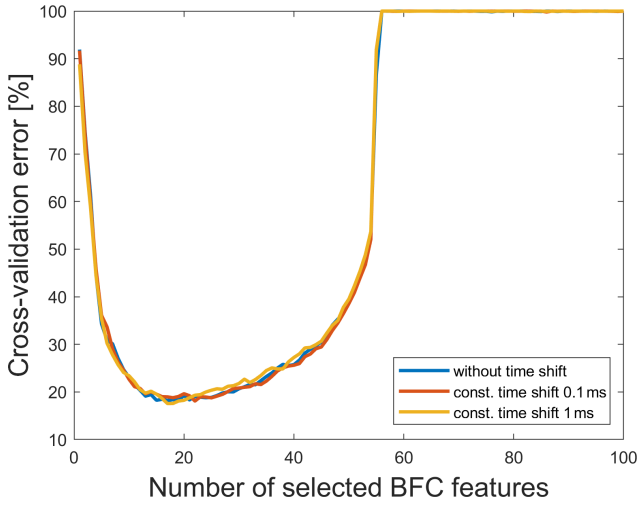


Figure 10. Cross-validation errors vs. the number of selected BFC features for constant shifted time windows with RFESVM as selector.

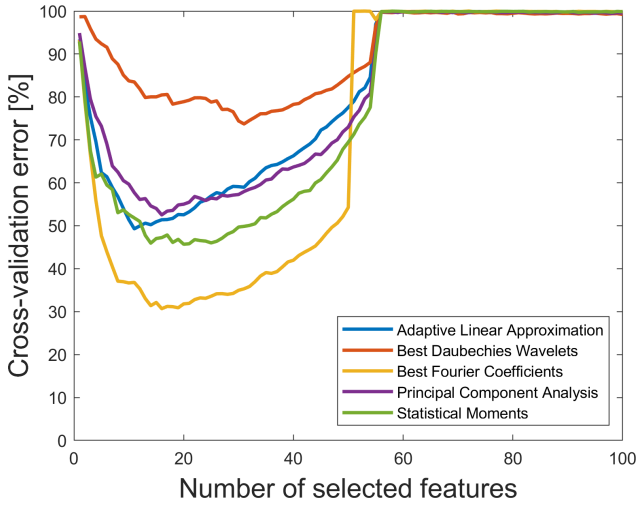


Figure 11. Cross-validation error vs. number of selected features for a maximum time shift of ±50 ms and RFESVM as selector. For a better visibility, only the results with RFESVM as selector are shown.

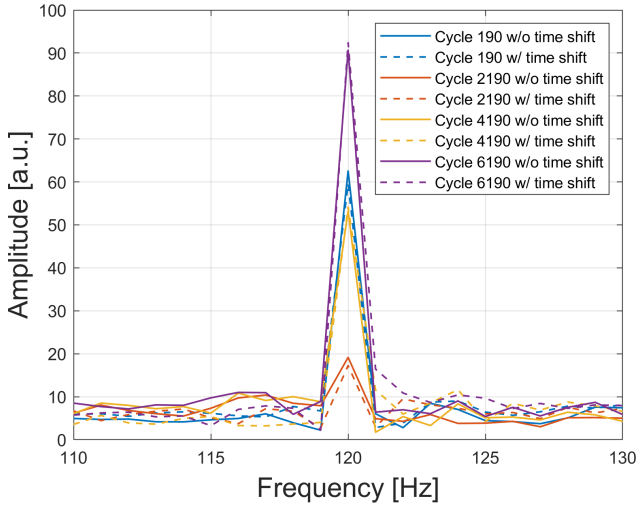


Figure 12. Best feature according to RFESVM (120 Hz of the active current) for the 2 kHz raw data set and the data set with random time shift of maximum 50 ms for three different cycles.

model to learn. For constant time shifts, on the other hand, the cross-validation error is nearly the same as for the raw data set (cf. Fig. 10), because every cycle is shifted by the same constant time, which does not affect the Fourier coefficients. Although, random time shifts have no influence on the amplitude spectrum in theory, but because of the experimental setup, there can occur cross-influences that make model building harder.

Since most of the results resulting from time-shifted data sets are almost equivalent to those obtained for the 2 kHz raw data set, not all results are explicitly discussed in this contribution. Only the data set with time shifts of maximum ±50 ms for all sensors' cycles is considered in more detail here. On the one hand, this time shift is the maximum possible when taking into account the cycle length of 2.8 s and evaluating a full second of the return stroke, and on the other hand, this time shift provides the worst cross-validation error for the combination of BFC and RFESVM. As shown in Fig. 11, the minimum cross-validation error is now 29.97 %, which is significantly worse than for the original data set without time shifts (18.18 %).

Figure 12 shows the frequency spectra for the 120 Hz feature of the active current (1 of the 17 most relevant features) for different cycles of the raw data set and the data set with random time shift of maximum 50 ms. It can be clearly seen

Table 3. Cross-validation error for the 2 kHz raw data set and 2 kHz data sets with different time shifts with BFC as extractor and RFESVM as selector.

Random time shift per cycle	Sensors with time shift	Min mean of 10-fold CV error	Selected features (thereof amplitudes)	Frequency range of selected features	Most relevant sensor (extracted features)
without	–	18.18 %	17 (71 %)	0–640 Hz	microphone, active current (each 29 %)
$\leq \pm 0.1$ ms	all	20.49 %	20 (90 %)	0–640 Hz	active current (35 %)
$\leq \pm 0.5$ ms	all	20.74 %	15 (93 %)	0–640 Hz	active current (27 %)
$\leq \pm 1$ ms	all	20.68 %	13 (100 %)	0–640 Hz	active current (23 %)
$\leq \pm 10$ ms	all	24.09 %	18 (100 %)	0–640 Hz	acceleration piston rod (22 %)
$\leq \pm 50$ ms	all	29.97 %	15 (100 %)	0–840 Hz	microphone, acceleration piston rod, acceleration ball bearing (each 20 %)

that this amplitude feature changes during the lifetime of the axis, but for different time-shifted data sets, it is nearly the same for the same cycle as for the raw data set. This is shown exemplary here with only one time-shifted data set.

For explanation of this behavior, let $x(t)$ denote the real-valued time domain signal for which information is available at discrete time points $t_0, \ldots, t_{N-1}$. The discrete Fourier transform (DFT) for the real-valued sequence $\boldsymbol{X} = (X_0, \ldots, X_{N-1})^\top$ is defined as

$$\hat{\boldsymbol{X}}_k = \sum_{n=0}^{N-1} \boldsymbol{X}_n \exp\left(-j\frac{2\pi n}{N}k\right) \text{ for } k = 0, \ldots, N-1. \quad (2)$$

If the DFT of the signal $x(t)$ is given by $\hat{\boldsymbol{X}}_k$, the DFT for the time-shifted signal $x(t-s)$ is given by

$$\hat{\boldsymbol{X}}_{k,\text{shifted}} = \hat{\boldsymbol{X}}_k \exp\left(-j\frac{2\pi n}{N}s\right) \text{ for } k = 0, \ldots, N-1. \quad (3)$$

The spectrum of the time-shifted signal is thus calculated from $\hat{\boldsymbol{X}}_k$, where each spectral component k experiences a frequency-proportional (linear) phase shift of $\exp\left(-j\frac{2\pi}{N}s\right)$. The amplitude spectrum of the time-shifted signal remains unchanged. Therefore, the amplitudes are robust against time shifts as seen in Fig. 12.

In industrial environments, there are often two different issues when using machine learning. First, there are synchronization problems within a sensor network which can be simulated here by training the model with the raw data set and applying the trained model on the data sets with different random time shifts. Figure 13 shows the classification error using a 10-fold cross-validation, which means the training per fold is carried out with 5663 random cycles of the 2 kHz raw data set; the remaining cycles of different data sets are used for the testing. It can be clearly seen that the classification error increases the larger the time shifts get. The classification error of 17.33 % is reached when applying the model only to the raw test data without time shifts. Applying the model built only with the raw data to time-shifted data with ± 0.1 ms already leads to a significant increase of the classification error (48.17 %). Thus, it is crucially important that the different sensors and cycles are synchronized. But when data are not well synchronized or if there is no information about the synchronization, the results can be improved somewhat by excluding the phase features, which can also be seen in Fig. 13. For the data set with ± 1 ms time shift, the result can be improved from 95.87 % using the model with amplitudes and the phases to 44.99 % when removing the phases out of the model.

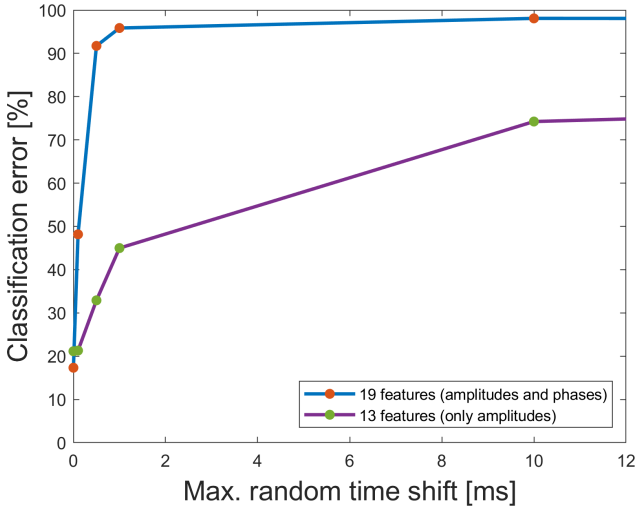


Figure 13. Classification error for one fold of the 10-fold cross validation using the raw data set for the model training and applying this model to data sets with different maximum random time shifts. Red dots represent models based on both amplitude and phase features, while green dots represent models using amplitude data only.

The second important issue is the choice of the time frame. Figure 14 shows that the time frame must be chosen exactly the same for all data sets, because the classification

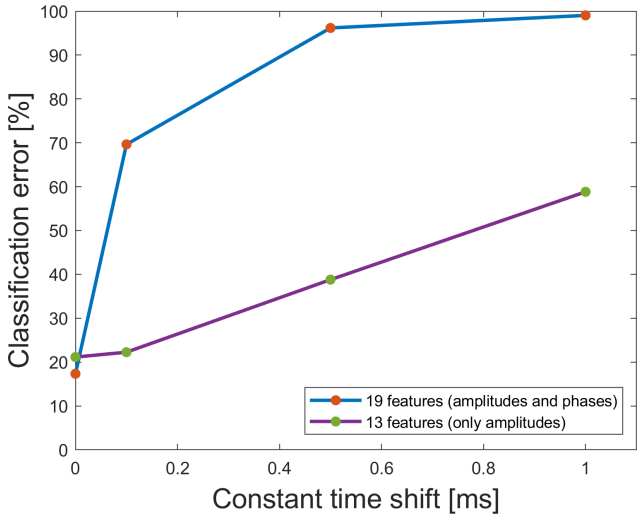


Figure 14. Classification error for one fold of the 10-fold cross validation using the 2 kHz raw data set for the model training and constant time-shifted data sets for the application of the trained model. Red dots represent models based on both amplitude and phase features, while green dots represent models using amplitude data only.

Table 4. Classification error for the prediction of the data set with 1 ms time shift by using different models.

Model/prediction	Without time shift	Without time shift, with 0.1 ms and 0.5 ms time shift
Amplitudes and phases	95.87 %	41.81 %
Only amplitudes	44.99 %	35.93 %

rate for one fold of the 10-fold cross-validation worsens from 17.33 %, applying the raw data for the testing, to 69.63 %, applying the data set with a time frame shifted by only 0.1 ms when using the model trained with the 2 kHz raw data set. In this case, it is also possible to improve the results by removing the phases from the model. For the data with the constant time shift of 0.1 ms, removing the phases and thus using only a model with amplitudes leads to a classification error of 22.26 % instead of 69.63 %.

A further improvement of the classification results can be achieved by training the model not only with the raw data but also with synthetically time-shifted data and considering only the amplitude features within the model (cf. Table 4).

To depict the effect of improving the classification error more clearly, the ±1 ms time-shifted data set is used for the testing of the model in all four cases in Fig. 15. Two different models are considered here. In the upper subfigures, the model was trained only with the 2 kHz raw data set, whereas in the lower ones the ±0.1 and ±0.5 ms time-shifted data are used for the model training in addition. The two subfigures on the left show the prediction of the lifetime with a resolution of 1 % when using the model, as it is resulting from the ML toolbox which means using both amplitudes and phases, whereas in the right ones only amplitudes are used. It can be clearly seen that the best classification error of 35.93 % for the ±1 ms time-shifted data set is reached with the model which is additionally trained with time shifts and consists of only amplitudes.

6 Conclusion and outlook

In this contribution, data sets with time synchronization errors were considered to investigate their influence on results obtained with a ML software toolbox for condition monitoring and fault diagnosis. Minimal synchronization errors between the individual sensors, when already present in the training data, only have a small effect on the cross-validation error achieved with the ML toolbox. However, if ML models are trained without any synchronization errors, applying these models to data sets even with minimal time shifts of 0.1 ms results in large classification errors, here for the prediction of the RUL of a critical component. This error can be reduced by modifying the feature extraction and excluding phase values after Fourier analysis in a first step. By adding artificially time-shifted data to the training set, a further improvement of the classification result is achieved. Thus, the study presented in this contribution provides important guidelines for improving the setup of distributed measurement systems, especially about the necessary synchronization between sensors. If no information about the synchronization within the network is available, it is suggested to generate artificially time-shifted data sets from the original data and use this extended data set for training the ML model. Note that this is similar to data augmentation suggested for improving the performance and robustness of neural networks (Wong et al., 2016).

It is also important to choose the time frame for the 1 s period correctly. Applying the model to data even with only a small shift of 0.1 ms of the time frame in comparison to the training data already leads to very poor classification results.

For future work, measurement uncertainty should be considered in addition to time synchronization errors as both contribute to data quality and are therefore expected to have a strong influence on ML results for condition monitoring or fault diagnosis. In the European research project “Metrology for the Factory of the Future” (Met4FoF), mathematical models for the consideration of metrological information in ML models are developed. For example, the project considers the classification within the ML toolbox by reviewing the robustness of the LDA as a classifier when using redundant features. Specifically, we will study how long the quality of the LDA results continues to improve with additional features and when the point is reached where the LDA fails, because the covariance matrix becomes singular; i.e., its determinant disappears.

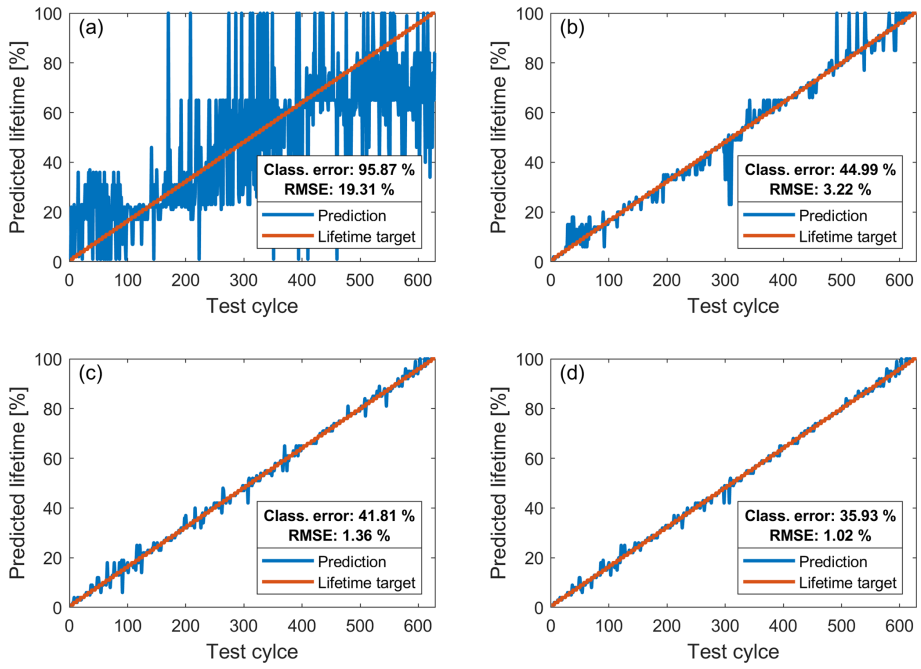


Figure 15. Predictions (blue) of the used EMC lifetime (steps of 1 %) for one fold of the 10-fold cross validation for the data set with time shifts of up to 1 ms and the assumed used lifetime target from 1 % to 100 % (red). **(a)** Model trained with raw data only using both amplitude and phase features. **(b)** Model trained with raw data only using only amplitude features. **(c)** Model trained with raw, 0.1, and 0.5 ms time-shifted data sets using both amplitude and phase features. **(d)** Model trained with raw, 0.1 and 0.5 ms time-shifted data sets only using amplitude features.

The current ML toolbox (see Fig. 3) does not take any measurement uncertainties into account. To overcome this limitation, the methods included in the toolbox are extended to allow for more robust and accurate failure analysis or condition monitoring applications such as predicting the RUL of components as discussed in this paper. The uncertainty evaluation for the BFC method was already presented by Eichstädt and Wilkens (2016). The uncertainty evaluation for ALA was recently published (Dorst et al., 2020). The uncertainty evaluation for the remaining three feature extraction methods is already developed and will be published soon. Thus, the ML toolbox can then provide features together with their uncertainty as determined from the uncertainty of the raw sensor data. Furthermore, the three feature selection algorithms can be replaced by filter-based selection algorithms which weight the features based on their uncertainties. Finally, the propagation of the uncertainty values through the LDA classifier is also completed. Thus, the extended ML toolbox, soon to be published, will be able to take the uncertainty of measured data into account to achieve improved models. In the future, we plan to add wrapper and embedded methods for the feature selection step of the ML toolbox that also consider uncertainties.

Code and data availability. The paper uses data obtained from a lifetime test of an EMC at the ZeMA test bed. As the full data set is confidential, a downsampled 2 kHz version of the data set is available on Zenodo https://doi.org/10.5281/zenodo.3929385 (Dorst, 2019).

The automated ML toolbox (Schneider et al., 2017, 2018b; Dorst et al., 2021a) includes all the code for data analysis associated with the current submission and is available at https://github.com/ZeMA-gGmbH/LMT-ML-Toolbox (last access: 23 August 2021) (Dorst et al., 2021b).

Author contributions. TD carried out the time shift analysis, visualized the results, and wrote the original draft of the paper. YR supported the data evaluation. TS developed the automated ML toolbox. SE and AS contributed with substantial revisions.

Competing interests. The authors declare that they have no conflict of interest.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Acknowledgements. The ML toolbox and the test bed were developed at ZeMA as part of the MoSeS-Pro research project funded by the German Federal Ministry of Education and Research in the call "Sensor-based electronic systems for applications for Industrie 4.0 – SElekt I 4.0", funding code 16ES0419K, within the framework of the German Hightech Strategy.

Financial support. Part of this work has received funding within the project 17IND12 Met4FoF from the EMPIR program co-financed by the Participating States and from the European Union's Horizon 2020 research and innovation program.

Review statement. This paper was edited by Ulrich Schmid and reviewed by two anonymous referees.

References

Beyer, K., Goldstein, J., Ramakrishnan, R., and Shaft, U.: When Is "Nearest Neighbor" Meaningful?, in: Database Theory – ICDT'99, Springer, Berlin, Heidelberg, 217–235, 1999.

Dorst, T.: Sensor data set of 3 electromechanical cylinder at ZeMA testbed (2 kHz), Zenodo [data set], https://doi.org/10.5281/zenodo.3929385, 2019.

Dorst, T., Ludwig, B., Eichstädt, S., Schneider, T., and Schütze, A.: Metrology for the factory of the future: towards a case study in condition monitoring, in: 2019 IEEE International Instrumentation and Measurement Technology Conference, Auckland, New Zealand, 439–443, https://doi.org/10.1109/I2MTC.2019.8826973, 2019.

Dorst, T., Eichstädt, S., Schneider, T., and Schütze, A.: Propagation of uncertainty for an Adaptive Linear Approximation algorithm, in: SMSI 2020 – Sensor and Measurement Science International, 366–367, https://doi.org/10.5162/SMSI2020/E2.3, 2020.

Dorst, T., Robin, Y., Schneider, T., and Schütze, A.: Automated ML Toolbox for Cyclic Sensor Data, in: MSMM 2021 – Mathematical and Statistical Methods for Metrology, 2021a.

Dorst, T., Robin, Y., Schneider, T., and Schütze, A.: Automated 35 ML Toolbox for Cyclic Sensor Data, in: MSMM 2021, Github [code], available at: https://github.com/ZeMA-gGmbH/LMT-ML-Toolbox (last access: 23 August 2021), 2021b.

Duda, R. O., Hart, P. E., and Stork, D. G.: Pattern Classification, in: A Wiley-Interscience publication, 2nd Edn., Wiley, New York, 2001.

Eichstädt, S.: Publishable Summary for 17IND12 Met4FoF "Metrology for the Factory of the Future", Zenodo [data set], https://doi.org/10.5281/zenodo.4267955, 2020.

Eichstädt, S. and Wilkens, V.: GUM2DFT – a software tool for uncertainty evaluation of transient signals in the frequency domain, Meas. Sci. Technol., 27, 055001, https://doi.org/10.1088/0957-0233/27/5/055001, 2016.

Guyon, I. and Elisseeff, A.: An Introduction to Variable and Feature Selection, J. Mach. Learn. Res., 3, 1157–1182, 2003.

Helwig, N.: Zustandsbewertung industrieller Prozesse mittels multivariater Sensordatenanalyse am Beispiel hydraulischer und elektromechanischer Antriebssysteme, PhD thesis, Dept. Systems Engineering, Saarland University, Saarbrücken, Germany, 2018.

Helwig, N., Schneider, T., and Schütze, A.: MoSeS-Pro: Modular sensor systems for real time process control and smart condition monitoring using XMR-technology, in: Proc. 14th Symposium Magnetoresistive Sensors and Magnetic Systems, 21–22 March 2017, Wetzlar, Germany, 2017.

Kohavi, R.: A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, in: Proceedings of the 14th International Joint Conference on Artificial Intelligence – Volume 2, IJCAI'95, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1137–1143, 1995.

Kononenko, I. and Hong, S. J.: Attribute selection for modelling, Future Generat. Comput. Syst., 13, 181–195, https://doi.org/10.1016/S0167-739X(97)81974-7, 1997.

Mörchen, F.: Time series feature extraction for data mining using DWT and DFT, Technical Report 33, Department of Mathematics and Computer Science, University of Marburg, Marburg, Germany, 1–31, 2003.

Olszewski, R. T., Maxion, R. A., and Siewiorek, D. P.: Generalized feature extraction for structural pattern recognition in time-series data, PhD thesis, Carnegie Mellon University, USA, 2001.

Rakotomamonjy, A.: Variable Selection Using SVM-based Criteria, J. Mach. Learn. Res., 3, 1357–1370, https://doi.org/10.1162/153244303322753706, 2003.

Robnik-Šikonja, M. and Kononenko, I.: Theoretical and Empirical Analysis of ReliefF and RReliefF, Mach. Learn., 53, 23–69, https://doi.org/10.1023/A:1025667309714, 2003.

Schneider, T., Helwig, N., and Schütze, A.: Automatic feature extraction and selection for classification of cyclical time series data, tm – Technisches Messen, 84, 198–206, https://doi.org/10.1515/teme-2016-0072, 2017.

Schneider, T., Helwig, N., Klein, S., and Schütze, A.: Influence of Sensor Network Sampling Rate on Multivariate Statistical Condition Monitoring of Industrial Machines and Processes, Proceedings, 2, 781, https://doi.org/10.3390/proceedings2130781, 2018a.

Schneider, T., Helwig, N., and Schütze, A.: Industrial condition monitoring with smart sensors using automated feature extraction and selection, Meas. Sci. Technol., 29, 094002, https://doi.org/10.1088/1361-6501/aad1d4, 2018b.

Schneider, T., Klein, S., Helwig, N., Schütze, A., Selke, M., Nienhaus, C., Laumann, D., Siegwart, M., and Kühn, K.: Big data analytics using automatic signal processing for condition monitoring | Big Data Analytik mit automatisierter Signalverarbeitung für Condition Monitoring, in: Sensoren und Messsysteme – Beitrage der 19. ITG/GMA-Fachtagung, 26–27 June 2018, Nürnberg, 259–262, 2018c.

Schütze, A., Helwig, N., and Schneider, T.: Sensors 4.0 – Smart sensors and measurement technology enable Industry 4.0, J. Sens. Sens. Syst., 7, 359–371, https://doi.org/10.5194/jsss-7-359-2018, 2018.

Sivrikaya, F. and Yener, B.: Time synchronization in sensor networks: a survey, IEEE Network, 18, 45–50, 2004.

Teh, H. Y., Kempa-Liehr, A. W., and Wang, K. I.-K.: Sensor data quality: a systematic review, J. Big Data, 7, 11, https://doi.org/10.1186/s40537-020-0285-1, 2020.

Tirado-Andrés, F. and Araujo, A.: Performance of clock sources and their influence on time synchronization in wireless

sensor networks, Int. J. Distrib. Sens. Netw., 15, 1–16, https://doi.org/10.1177/1550147719879372, 2019.

Usuga Cadavid, J. P., Lamouri, S., Grabot, B., Pellerin, R., and Fortin, A.: Machine learning applied in production planning and control: a state-of-the-art in the era of industry 4.0, J. Intel. Manufact., 31, 1531–1558, https://doi.org/10.1007/s10845-019-01531-7, 2020.

Wold, S., Esbensen, K., and Geladi, P.: Principal component analysis, Chemometr. Intel. Labor. Syst., 2, 37–52, https://doi.org/10.1016/0169-7439(87)80084-9, 1987.

Wong, S. C., Gatt, A., Stamatescu, V., and McDonnell, M. D.: Understanding Data Augmentation for Classification: When to Warp?, in: 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA), 30 November–2 December 2016, Gold Coast, QLD, Australia, 1–6, https://doi.org/10.1109/DICTA.2016.7797091, 2016.

Yiğitler, H., Badihi, B., and Jäntti, R.: Overview of Time Synchronization for IoT Deployments: Clock Discipline Algorithms and Protocols, Sensors, 20, 5928, https://doi.org/10.3390/s20205928, 2020.

6.3 Paper E: Towards Interpretable Machine Learning for Automated Damage Detection Based on Ultrasonic Guided Waves

Since the algorithmic approach for ML-based condition monitoring assumes the input data to be a time series of equal length, the same approach can be applied to a different field with a similar signal structure. The following paper shows the application of the automated machine learning toolbox to structural health monitoring. It shows the exemplary application of ultrasonic-guided waves-based damage detection in a composite plate. This work formed the foundation for the project KI-Mono, which will further investigate the use of machine learning for structural health monitoring with ultrasonic-guided waves.

This example also shows the combination of the automated machine learning toolbox's purely statistical approach with domain knowledge to perform suitable preprocessing for temperature compensation using differential measuring with optimal baseline selection (OBS) and baseline signal stretch (BSS).

The algorithms applied are the automated machine learning toolbox plus SVM for classification. Additionally, it is used to define dataset splits for group-based cross-validation that quantify robustness against variations in temperature and fault location. The stepwise approach with feature extraction and selection also allows physical interpretation of the most important features used for fault detection.

Towards Interpretable Machine Learning for Automated Damage Detection Based on Ultrasonic Guided Waves

Christopher Schnur[1], Payman Goodarzi[1], Yevgeniya Lugovtsova[2], Jannis Bulling[2], Jens Prager[2], Kilian Tschöke[3], Jochen Moll[4], Andreas Schütze[1,5], and Tizian Schneider[1,5]

[1]Lab for Measurement Technology, Saarland University, Saarbrücken, Germany

[2]Department of Non-Destructive Testing, Acoustic and Electromagnetic Methods Division, Bundesanstalt für Materialforschung und -Prüfung (BAM), Berlin, Germany

[3]Systems for Condition Monitoring, Fraunhofer Institute for Ceramic Technologies and Systems IKTS, Dresden, Germany

[4]Department of Physics, Goethe University Frankfurt, Frankfurt, Germany;

[5]Research Group Data Engineering and Smart Sensors, ZeMA - Centre for Mechatronics and Automation Technology gGmbH, Saarbrücken, Germany

Sensors (2022), 22 (1), 406

The original paper can be found online at https://doi.org/10.3390/s22010406.

© 2022 by the authors. This open-access article is distributed under the terms and conditions of the Creative Commons Attribution 4.0 License. (http://creativecommons.org/licenses/by/4.0/).

sensors

MDPI

Article

Towards Interpretable Machine Learning for Automated Damage Detection Based on Ultrasonic Guided Waves

Christopher Schnur [1,*], Payman Goodarzi [1], Yevgeniya Lugovtsova [2], Jannis Bulling [2], Jens Prager [2], Kilian Tschöke [3], Jochen Moll [4], Andreas Schütze [1,5] and Tizian Schneider [1,5]

1 Lab for Measurement Technology, Saarland University, 66123 Saarbrücken, Germany; p.goodarzi@lmt.uni-saarland.de (P.G.); schuetze@lmt.uni-saarland.de (A.S.); t.schneider@lmt.uni-saarland.de (T.S.)
2 Department of Non-Destructive Testing, Acoustic and Electromagnetic Methods Division, Bundesanstalt für Materialforschung und -Prüfung (BAM), 12205 Berlin, Germany; yevgeniya.lugovtsova@bam.de (Y.L.); jannis.bulling@bam.de (J.B.); jens.prager@bam.de (J.P.)
3 Systems for Condition Monitoring, Fraunhofer Institute for Ceramic Technologies and Systems IKTS, 01109 Dresden, Germany; kilian.tschoeke@ikts.fraunhofer.de
4 Department of Physics, Goethe University Frankfurt, 60438 Frankfurt, Germany; moll@physik.uni-frankfurt.de
5 Research Group Data Engineering and Smart Sensors, ZeMA—Center for Mechatronics and Automation Technology gGmbH, 66121 Saarbrücken, Germany
* Correspondence: c.schnur@lmt.uni-saarland.de

Abstract: Data-driven analysis for damage assessment has a large potential in structural health monitoring (SHM) systems, where sensors are permanently attached to the structure, enabling continuous and frequent measurements. In this contribution, we propose a machine learning (ML) approach for automated damage detection, based on an ML toolbox for industrial condition monitoring. The toolbox combines multiple complementary algorithms for feature extraction and selection and automatically chooses the best combination of methods for the dataset at hand. Here, this toolbox is applied to a guided wave-based SHM dataset for varying temperatures and damage locations, which is freely available on the Open Guided Waves platform. A classification rate of 96.2% is achieved, demonstrating reliable and automated damage detection. Moreover, the ability of the ML model to identify a damaged structure at untrained damage locations and temperatures is demonstrated.

Keywords: composite structures; structural health monitoring; carbon fibre-reinforced plastic; interpretable machine learning; automotive industry

check for updates

Citation: Schnur, C.; Goodarzi, P.; Lugovtsova, Y.; Bulling, J.; Prager, J.; Tschöke, K.; Moll, J.; Schütze, A.; Schneider, T. Towards Interpretable Machine Learning for Automated Damage Detection Based on Ultrasonic Guided Waves. *Sensors* **2022**, *22*, 406. https://doi.org/10.3390/s22010406

Academic Editor: Branko Glisic

Received: 4 November 2021
Accepted: 29 December 2021
Published: 5 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Machine learning (ML) techniques require a large number of measurements for adequate training and reliable decision-making. Therefore, ML is well suited for structural health monitoring (SHM) applications in which one or multiple sensors are permanently attached to the structure so that structural measurements can be recorded frequently. This rich data pool can be exploited by ML techniques to train a model that can detect damages or anomalies, allowing for fully automated damage detection.

Several ML methods have been developed in the last few years to solve various SHM and damage detection problems, especially by using neural networks (NN) [1–5]. Even though ML methods are already well established in vibration-based SHM [6], their use in guided wave-based SHM is currently rising [7–9]. For instance, Roy et al. [7] described an unsupervised learning approach for structural damage identification under varying temperatures based on an NN. Their methodology is validated with measurements from coupon samples in a uniaxial testing machine. More recently, Miorelli et al. [8] demonstrated that support vector machines (SVM) trained on numerical data can be used to solve the inverse problem for damage detection and sizing from experimental

guided wave (GW) images. They used a circular array of transducers on an isotropic metal plate with through-holes of different sizes modelled at different locations. Mariani et al. [9] showed improvements in automatic damage detection when using a causal dilated convolutional NN without the need for feature engineering by a human operator. Qiu [1] studied Gaussian mixture models for GW in SHM systems using measurements from a full-scale fatigue test.

Keogh et al. [10] found, in a study of 340 papers, that new methods are tested on average on 1.3 different datasets and compared to 0.9 other methods only, and routine applications are desired to reduce the requirement for data scientists to adapt the ML methods for industrial applications. The contribution of this work is therefore the adaptation and application of an existing ML framework previously used for condition monitoring of industrial machines to GW-based SHM to enable autonomous damage detection. The framework is based on a toolbox combining multiple established ML algorithms that was successfully applied to various other datasets (cf. [11]). A reference dataset from the Open Guided Waves platform is used in this work, consisting of GW measurements performed with an array of piezoelectric transducers on a carbon fibre-reinforced polymer (CFRP) under varying temperatures [12]. The analysis shows that a classification rate of 96.2% can be achieved, demonstrating reliable and automated damage detection. Moreover, the ability of the ML model to detect damages at untrained damage locations and temperatures outside of the trained temperature range is also demonstrated. The methodology presented in this manuscript should be seen as a general pathfinder rather than a tailored solution.

Neural networks are commonly used in SHM applications but are difficult to interpret, and therefore their use in safety-relevant applications is limited. The methodology presented in this paper focuses on interpretability, meaning that the ML results must be physically interpretable to enable the use of ML also in safety-relevant applications. We compare our methodology against the performance of an NN applied on the same dataset [9]. Furthermore, we demonstrate that our ML methodology enables a straightforward learning procedure without the need for domain-specific knowledge and highly educated staff like data scientists, which is very important for wider application of these methods in the industry. On the other hand, it must be noted that even better performance can be achieved with domain-specific knowledge by highly educated staff.

The outline of the paper is as follows. First, the experimental setup along with the pre-processing of signals for temperature compensation is presented, followed by the description of the automated toolbox. Next, the performance of the automated ML framework is analysed. To do so, a realistic validation scenario is chosen, which is a crucial step to minimise overfitting. In addition, the selection of the hyper-parameters is motivated to achieve a higher performance. The Results section first provides a visualisation of the data using principal component analysis. Then, the performance of different algorithms for automated damage detection is presented and discussed. Moreover, the robustness of the algorithms against different damage locations and temperatures is tested and a comparison to results achieved with a deep learning NN by Mariani et al. [9] is presented. The paper closes with conclusions and the outlook.

2. Machine Learning Approach

2.1. Description of the Experimental Setup

This study is based on a freely available benchmark dataset for guided wave-based SHM with varying temperatures, recorded by Moll et al. [12]. Here, multiple ultrasonic transducers (T_1–T_{12}) were attached to a carbon fibre-reinforced plastic (CFRP) plate, as well as a sequentially added detachable mass (aluminium disc) at four different locations (D_{04}, D_{12}, D_{16}, D_{24}) to simulate structural damages. The impact of the simulated damages on the measurements can be considered a rough approximation of real delamination (e.g., decrease in amplitude and changes in time of flight) [12]. The exact positions of the transducers and the damage locations as well as their distance to the direct signal path (T_4 to T_9) can be found in Table 1. Note that, in the scope of this manuscript, the term "simulated

damage" denotes an experimental simulation of a damaged material and does not refer to numerical simulation.

Table 1. Position of the transducers and the damage locations [12]. The distance of the damage locations to the direct signal path had been calculated.

Label	Position on x-Axis (mm)	Position on y-Axis (mm)	Distance to Signal Path (mm)
	Transducer positions		
Transducer 4	210	470	0
Transducer 9	290	30	0
	Damage positions		
Damage 04	65	400	155
Damage 12	195	330	40
Damage 16	335	260	85
Damage 24	450	190	186

A schematic of the CFRP plate with the positions of the transducers and damages is shown in Figure 1a. The subsequent analysis considers the case of a 40 kHz Hann-windowed tone-burst signal with five cycles (Figure 1b) sent by T_4 and received by T_9 for all four damage locations D_{04}, D_{12}, D_{16}, and D_{24} as well the undamaged structure. Each measurement contains only one simulated damage at a time. During the experiment, the plate was subjected to several temperature cycles between 20 and 60 °C in a climatic chamber (Figure 1c) at constant humidity (50% RH, mean: ~50.1%, standard deviation ~0.3%). For studies concerning the impact of humidity on CFRP the reader is referred to Schubert et al. [13]. Note that measurements for the undamaged plate were performed on two temperature cycles instead of only one. For the pre-processing (Section 2.2) the ascending flank (20 °C to 60 °C in 0.5 °C steps) of the first temperature cycle of the undamaged plate was used as a database (DB, Figure 1c) for the optimal baseline selection (OBS) of reference signals (cf. Section 2.2), and the descending flank is labelled "undamaged group 1" (UG_1). The second temperature cycle (ascending and descending flank) is labelled "undamaged group 2" (UG_2). These two different groups are later used in the validation (Section 2.4).

Multiple configurations were analysed and two representative scenarios chosen, one where the transducers were located in the middle of the CFRP plate (T_4 and T_9) and the other where they were located at the edge (T_1 and T_7; Section 3.3). In the scope of this study, we focused on one transducer combination at a time to be able to interpret the ML results more easily and, more importantly, to reduce the complexity and cost of later SHM configurations. Although the performance could be increased by using the information of all sensors, the aim of this study was to gain a better understanding of which configuration is necessary to reliably detect a damaged structure.

2.2. Signal Pre-Processing

Increasing the temperature of the CFRP decreases the phase and group velocity of guided wave modes and increases material attenuation. Unsupervised principal component analysis (PCA) on the raw data identifies this effect to be by far the most dominant variation in the dataset (Appendix A, Figure A1). It masks less significant fault symptoms that indicate a damage in the CFRP specimen. This may cause the unsupervised and automated feature extraction strategy described below to miss these symptoms. To mitigate this effect, differential measurement techniques—optimal baseline selection (OBS) and baseline signal stretch (BSS)—were employed for temperature compensation [14]. This approach is schematically shown in Figure 2 and comprises the following steps:

OBS is applied, where the measured signal is compared to all signals of the reference database from the intact structure covering the full experimental temperature range. The closest match (reference signal) as determined by the root mean square error (RMSE) is chosen as the optimal baseline.

BSS is applied on the baseline signal:

a. The baseline signal is stretched on the time axis to best fit the measured signal, again as determined by the RMSE.
b. The stretched baseline is shifted on the time axis to achieve the best fit to the measured signal in terms of RMSE.
c. The shifted baseline's amplitude is scaled to match the measured signal in terms of RMSE.

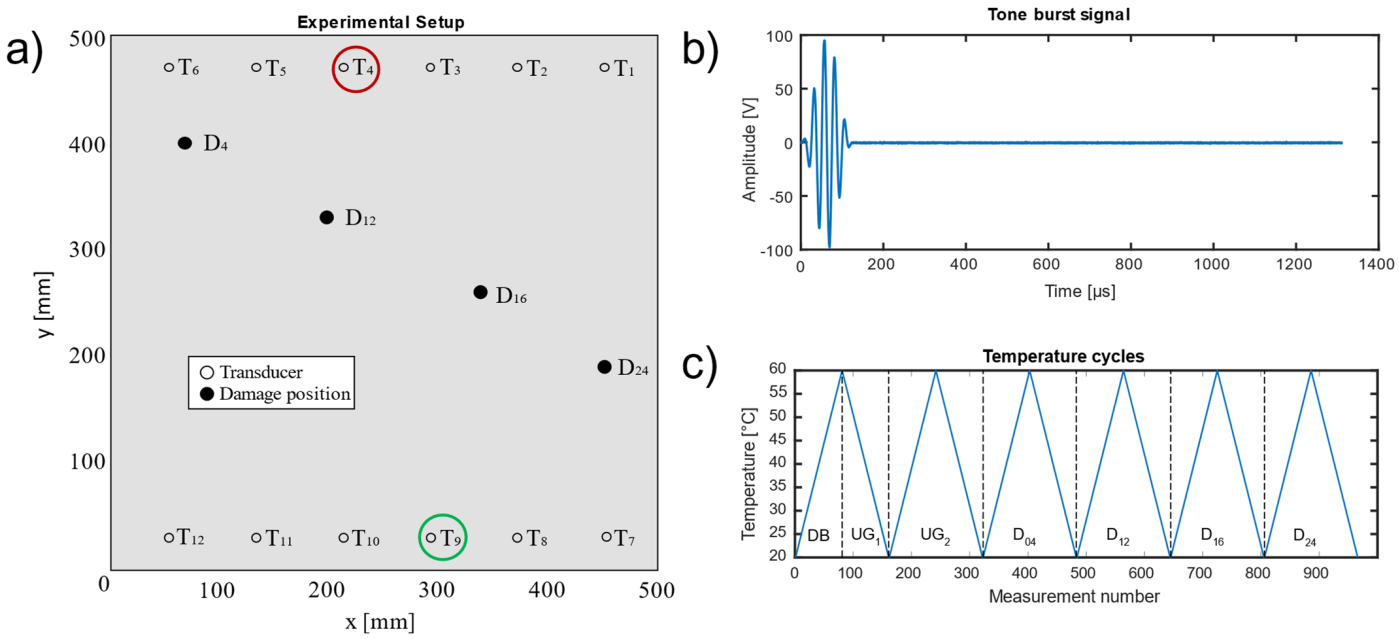


Figure 1. (**a**) Schematic of the experimental setup [12]. The analysed sensor combination is indicated by circles (the red circle indicates the transmitter T4, whereas the green circle indicates the receiver T9). The considered damage positions (D_{04}, D_{12}, D_{16}, D_{24}) are indicated by filled black dots. (**b**) 40 kHz Hann-windowed tone-burst signal with five cycles. (**c**) Temperature of the climatic chamber for each measurement number, where the dotted lines indicate the corresponding groups of the database, undamaged and damaged measurements.

This modified baseline is subtracted from the measured signal to obtain the difference (residual) signal.

All approaches, methods, and results reported below are based on the signals taken from the reference database being pre-processed using OBS and BSS algorithms.

The database in this study contained 81 measurements with only one measurement per 0.5 °C temperature step (cf. Section 2.1). Here, we selected the minimum database that contained all temperatures to keep the computation time low, since OBS compares measured signals to each signal in the database. In real-life SHM applications, the number of measurements of an intact structure could be much higher by adding every new measurement (of an intact structure) to the database, rapidly increasing its size. However, we suggest focussing on the composition of the database rather than its size because a database representing a high variance of, e.g., environmental conditions like temperature, humidity, etc., should increase the robustness of the ML model.

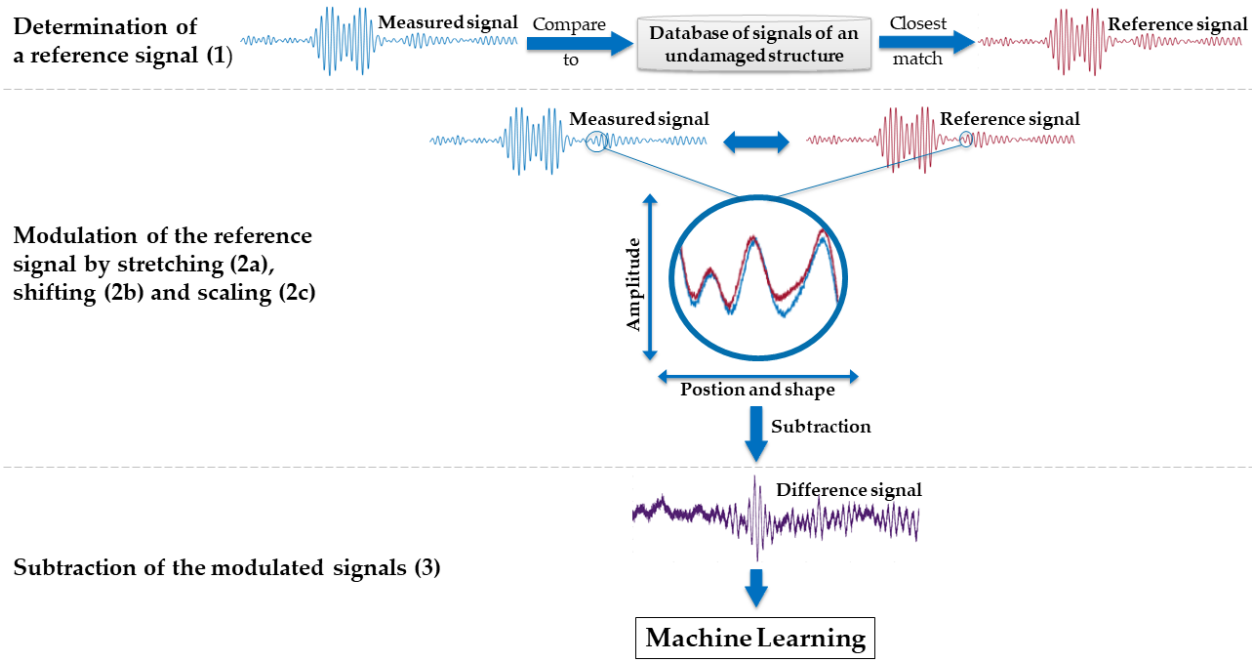


Figure 2. Pre-processing of the raw data to compensate for temperature-related effects by using optimal baseline selection (OBS) and baseline signal stretch (BSS).

2.3. Automated Toolbox

Signal classification was performed using a fully automated toolbox for industrial time series feature extraction and selection [15]. All algorithms are part of the MATLAB-based open-source Automated ML Toolbox for Cyclic Sensor Data [16] and its compiled version DAV^{3}E—Data Analysis and Verification/Visualisation/Validation Environment [17] (Supplementary Materials), both developed by the Lab for Measurement Technology at Saarland University. This automated toolbox combines five unsupervised and complementary feature extraction (FE) methods with three complementary methods for feature selection (FS) (Table 2).

Table 2. Feature extraction and selection methods of the automated toolbox [17].

Methods	Abbreviation	Literature
Feature Extraction Methods		
Adaptive linear approximation	ALA	[18]
Principal component analysis	PCA	[19]
Best Fourier coefficients	BFC	[20]
Best Daubechies wavelets	BDW	[21]
Statistical moments	SM	[22]
Feature Selection Methods		
Recursive feature elimination support vector machines *	RFE-SVM	[23,24]
RELIEFF *	RELIEFF	[25,26]
Pearson correlation coefficient	PCC	[27]

* Before this feature selection method is applied, the number of features is reduced to 500 in a first feature selection step based on the Pearson correlation coefficient.

To keep the computation within a reasonable time, the extracted number of features was reduced in a first feature (pre-)selection to the 500 features with the highest PCC. Thus, 15 FE/FS combinations were automatically analysed within the toolbox, using a simple classification approach based on supervised linear discriminant analysis (LDA) with Mahalanobis distance classification [28]. Out of the 15 combinations, the best FE/FS combination was automatically selected based on the highest test accuracy using 10-fold cross-validation.

If needed, this approach can be extended using more sophisticated classification algorithms. In this study, further investigations with a support vector machine (SVM) with a radial basis function kernel (RBF-Kernel) were performed, because this classifier achieved the best performance (highest accuracy in the shortest time) in a comparison of 14 different families of classification algorithms on 115 binary datasets [29]. Other relevant examples of using SVM in the context of SHM can be found in [6,8].

2.4. Validation Scenario

In real-world applications, the exact position of damage is unknown and generally differs from simulated or trained ones. Therefore, damage detection is required to also detect damages located at positions that were not included in the training data by learning certain global damage characteristics that are robust against changes in damage location. Thus, the model is trained with the pre-processed data as a binary decision (damaged/undamaged). The standard stratified 10-fold cross-validation (Figure 3, left) divides the dataset into 10 sub-datasets (folds), where each fold has the same proportion of damaged and undamaged data. Here, simple ML approaches can achieve a high accuracy on the Open Guided Wave data, which shows statistical significance but not the needed robustness against untrained damage positions, since all simulated damages (D_{04}, D_{12}, D_{16}, D_{24}) are included in each training set. Stratified CV cannot guarantee that the model learns general characteristics of a damaged or undamaged structure instead of only damage-specific and position-related characteristics, which only occur at the locations of the trained damages. This may result in overfitting, meaning that the ML model is trained only for specific damage locations and is then unable to identify damages at other locations. Therefore, 10-fold cross-validation is replaced by leave-one-group-out cross-validation (LOGOCV; Figure 3, right). To do so, the dataset is divided into data subsets with respect to the corresponding groups (UG_1, UG_2, D_{04}, D_{12}, D_{16}, D_{24}), allowing for the exclusion of each damage location from the training data once and thus making this damage location completely unknown to the ML model. The excluded group is then used to validate the performance of the trained model. To ensure that the training dataset always contains data of the undamaged sample, these measurements are split into two groups (UG_1, UG_2).

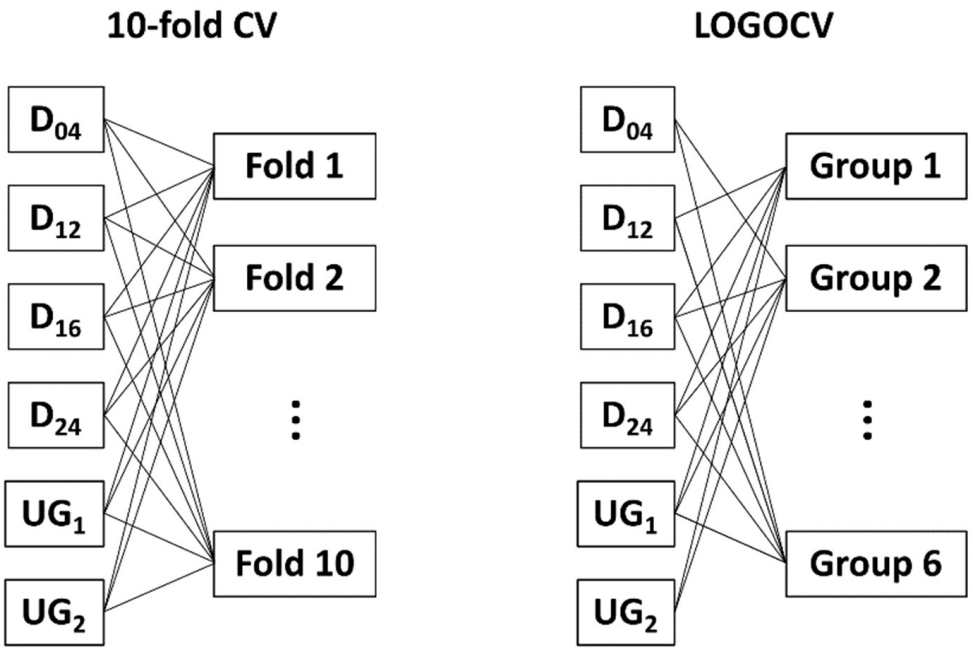


Figure 3. Comparison of 10-fold CV (**left**) and LOGOCV (**right**).

The flowchart of this methodology is depicted in Figure 4. It shows how the sensor signals are used for the training and automated algorithm selection. After selecting the

best FE method in combination with the chosen robust feature selection (RELIEFF) and classification (SVM with RBF kernel) based on testing with LOGOCV, the model is trained with all available data. It is then applied to new measurements, classifying them as either damaged or undamaged.

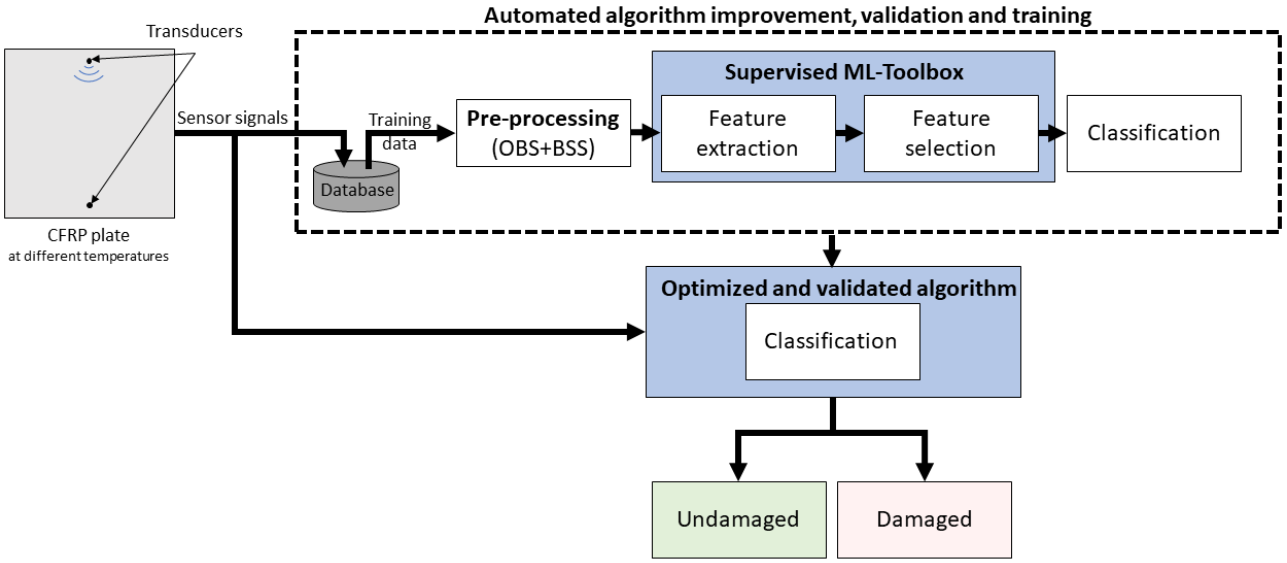


Figure 4. Flowchart illustrating the ML framework and the decision-making strategy.

2.5. Hyper-Parameter Selection

To increase the performance of the ML model, a selection of the hyper-parameters *C* (regularisation parameter of the SVM) and the number of features was performed. Here, a grid search approach was used based on Gui et al., who tested three methods for SVM optimisation in SHM for damage detection with a grid search, achieving the highest accuracy [30]. In this approach, an ML model is trained and validated with every possible combination of hyper-parameters in a pre-defined range. The combination with the highest validation accuracy is chosen and finally tested with independent data not included in the training and validation data.

Table 3 shows the values and tested number of values for each parameter. To reduce computational time and resources while still covering a broad range of values, the step size for the number of features increased the higher it became. The maximum number of features was set to 500 based on the feature pre-selection, which reduced the number of extracted features to 500 to avoid overfitting. Similarly, to cover a wide range of values for the regularisation parameter *C*, logarithmic scaling was chosen, i.e., $C = 10^{0.5i}$, $i \, \epsilon \, (-2, 8)$.

Table 3. Parameters and values used for the grid search approach to improve the ML model. "Number of features" means the selected features that are used for classification. Bold numbers indicate the selected hyper-parameters for Section 3.5.

Hyper-Parameter	# of Values	Values
Number of features	31	1, 2, …, 10, 15, 20, …, **25**, …, 50, 60, 70, …, 100, 150, …, 500
Regularisation parameter *C*	11	0.1, **0.3**, 1, 3.2, 10, 31.6, 100, 316.2, 1000, 3162.3, 10,000

Note that the parameter σ of Equation (A5) (cf. Appendix B) was not part of the grid search, as it is automatically optimised by MATLAB. After performing the grid search approach, the algorithm selects a parameter combination achieving high accuracy while using as few features as possible. Regarding the regularisation parameter *C*, if multiple parameter combinations achieve maximum accuracy, a trade-off can be made. Whereas a larger value for *C* suppresses misclassifications, a smaller value for *C* allows misclassifications to a certain degree [31]. Here, we preferred a smaller value for *C* to achieve a higher tolerance

for misclassifications and higher robustness against outliers [31]. Further information on the theoretical background of SVMs can be found in [31,32] on the difference between hyper-parameter tuning as performed here and hyper-parameter optimisation of SVMs as described in [33–35].

3. Results and Discussion

3.1. Principle Component Analysis

Principal component analysis is a common unsupervised method for visualising data to gain a better understanding of the nature of the dataset. Figure 5a shows the result of the scatterplots of the first five principal components (PC) based on the pre-processed data, with the corresponding variance that each principal component explains and the histograms on the diagonal. Here, the second and third PC (PC2, PC3), indicated by a red box, showed better separability than the remaining PCs. Note that PCA is used here for visualisation of the pre-processed data (OBS + BSS) only, without any additional data treatment.

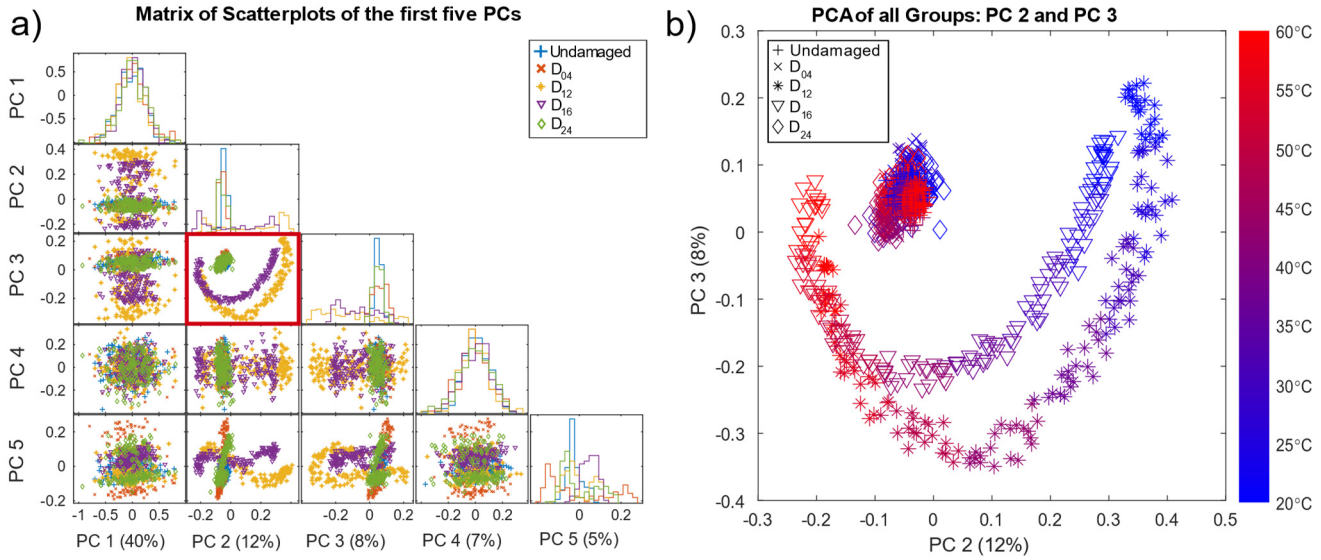


Figure 5. (**a**) Matrix of the first five PCs of the PCA on the pre-processed data (undamaged plate and all simulated damage locations) with their histograms on the diagonal and the variance explained by each PC given as a percentage in brackets. The red box indicating the scatterplot of PC 2 and PC 3 is also shown in (**b**), where the data points are additionally coloured by their corresponding temperature.

The scatter plot of PC 2 and PC 3 (Figure 5b) reveals good separability for damage locations D_{12} and D_{16} located in the direct signal path between T4 and T9, where waves reflected from and transmitted through the damage (resulting in decreased amplitudes) had a higher impact on the measurements. Since D_{04} and D_{24} were not in the direct signal path, their influence on the received signal was smaller. D_{04}, D_{24}, and the undamaged data formed a cluster in the centre. In addition, Figure 5b shows all pre-processed measurements coloured by the corresponding temperature. Thus, the crescent-moon shape of the signals for D_{12} and D_{16} was mainly due to the temperature effect, which was not fully compensated by the OBS + BSS pre-processing. Figure 5b implies that measurements of D_{12} and D_{16} at higher temperatures were more difficult to discriminate, as they lay closer to each other as well as to the cluster of the undamaged plate and damages D_{04} and D_{24}.

These plots also show that pre-processing can, at least to a certain degree, suppress temperature effects and highlight damage symptoms. However, the damage cases D_{04} and D_{24} overlapped with the undamaged data UG_1 and UG_2 in the first five PCs, which explains 72% of the variance.

3.2. Results of the Automated Toolbox and Improvement of the Algorithms

In the following, we describe our approach to find a robust model with a high classification rate. When using the standard classifier of the toolbox, the highest resulting test accuracy was 88%, achieved using BFC as a feature extractor and RFE-SVM for feature selection (Table 4). This classification rate is inadequate, especially for safety-relevant applications. Table 4 provides further information on how the different FE/FS combinations performed. Here, a user of the toolbox could see that, besides the expected BFC extractor, the SM extractor might be interesting for further analysis, whereas, e.g., ALA is not suitable for FE here.

Table 4. Overview of the testing accuracies of all 15 combinations of the automated toolbox, derived in a previous study [36]. The highest testing accuracy is shown in bold.

	Testing Accuracy for Each Algorithm Combination of the Automated Toolbox				
	PCA	**BFC**	**BDW**	**ALA**	**SM**
Pearson	42%	73%	42%	31%	81%
RELIEFF	42%	80%	43%	31%	78%
RFE-SVM	52%	**88%**	48%	31%	81%

To increase the performance, the feature extraction method was improved, and the feature selection and classification methods were replaced. Due to the relatively high robustness against incomplete and noisy data in real-life applications, RELIEFF was chosen as the feature selection algorithm [25,26]. As a classifier, SVM with RBF kernel was chosen due to its good performance in a comparison of 14 families of classification algorithms on 115 binary datasets [19].

The BFC extractor of the toolbox initially extracted 5% (1310 features) of the frequency spectrum by ranking them according to the highest amplitude, and extracted those frequencies and their corresponding phase angles. This value was increased up to 10% (2620 features) to also consider features with a lower signal amplitude in the training. To achieve a reasonable computing time, the resulting 2620 features were first reduced to 500 by selecting the features with the highest Pearson correlation to the damage. The final FS method, RELIEFF, reduced the number of features down to 20. This number of features was determined by averaging the obtained feature numbers of the six models in the grid search. This improvement of the toolbox resulted in a damage classification rate of 96.2% (Table 5) compared to 88%, i.e., reducing the number of misclassified measurements from 118 to 33. A detailed description of the improved algorithms and the procedure is given in Appendix B.

Table 5. Overview of the testing accuracy and number of misclassifications of the improved algorithms (BFC, RELIEFF with Pearson pre-selection, RFE-SVM) of the toolbox for GW-based SHM.

	Results of the Improved Algorithms of the Toolbox						
Damage Case	**UG_1**	**UG_2**	**D_{04}**	**D_{12}**	**D_{16}**	**D_{24}**	**Total**
Number of samples	80	161	161	161	161	161	885
Misclassifications	1	3	0	0	0	29	33
Accuracy	98.7%	98.1%	100%	100%	100%	82.0%	96.2%

It is worth mentioning that due to the validation strategy (LOGOCV), these results are robust for temperature variations as well as damages at unknown positions. The corresponding predictions are shown in Figure 6. Note that most misclassifications occurred for measurements of damage at position D_{24}, which is the location farthest from the direct path in this study (186 mm; Table 1), in combination with high temperatures (>45 °C).

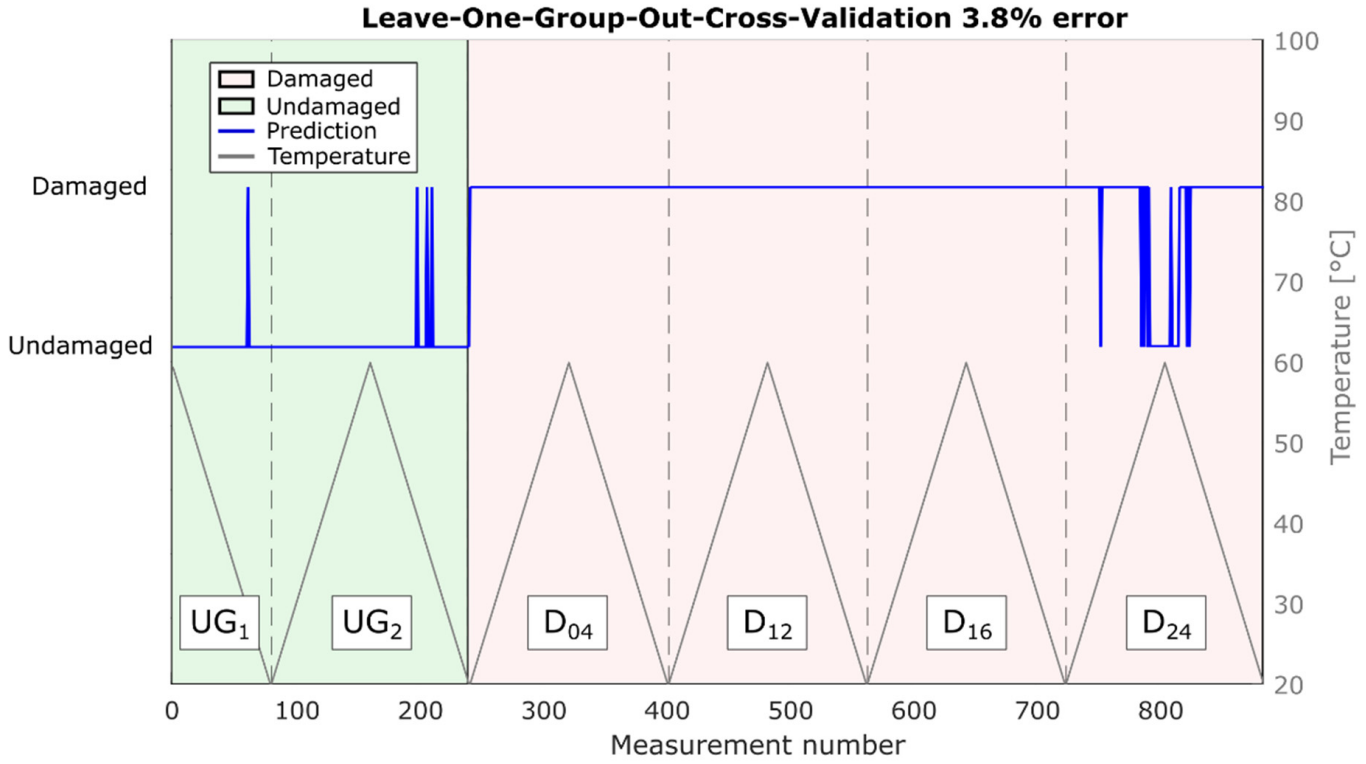


Figure 6. Damage classification results of the leave-one-class-out cross-validation. The plot is divided into six sections by dotted lines. Each section represents a heat cycle with one specific damage condition (undamaged and damaged D_{04}, D_{12}, D_{16}, D_{24}).

With the proposed transparent FE/FS approach, the ranking of the features that are most often selected for damage detection can help with a physical interpretation. The five highest ranks (eight features) are listed in Table 6.

Table 6. Ranked BFC features, i.e., frequencies, for transducer combinations 4 and 9 with their rank, total selections, amplitude selections, and phase selections. Ranking is based on how often the respective frequency is selected either as an amplitude or a phase feature in the six different LOGOCV models. Four frequencies are selected six times each.

		Ranked Frequencies (BFC Features)			
Nr.	**Rank**	**Frequency**	**Total Selections**	**Amplitude Selections**	**Phase Selections**
1	1	38.9 kHz	10	4	6
2	2	42.7 kHz	9	6	3
3	3	45.0 kHz	8	3	5
4	4	35.9 kHz	7	2	5
5	5	27.5 kHz	6	6	0
6	5	36.6 kHz	6	5	1
7	5	42.0 kHz	6	0	6
8	5	45.8 kHz	6	3	3

These frequencies were all included in the frequency spectrum of the Hann-windowed excitation frequency, as shown in Figure 7, indicating that they were not a misinterpretation of environmental influences but indeed originated from the excitation signal.

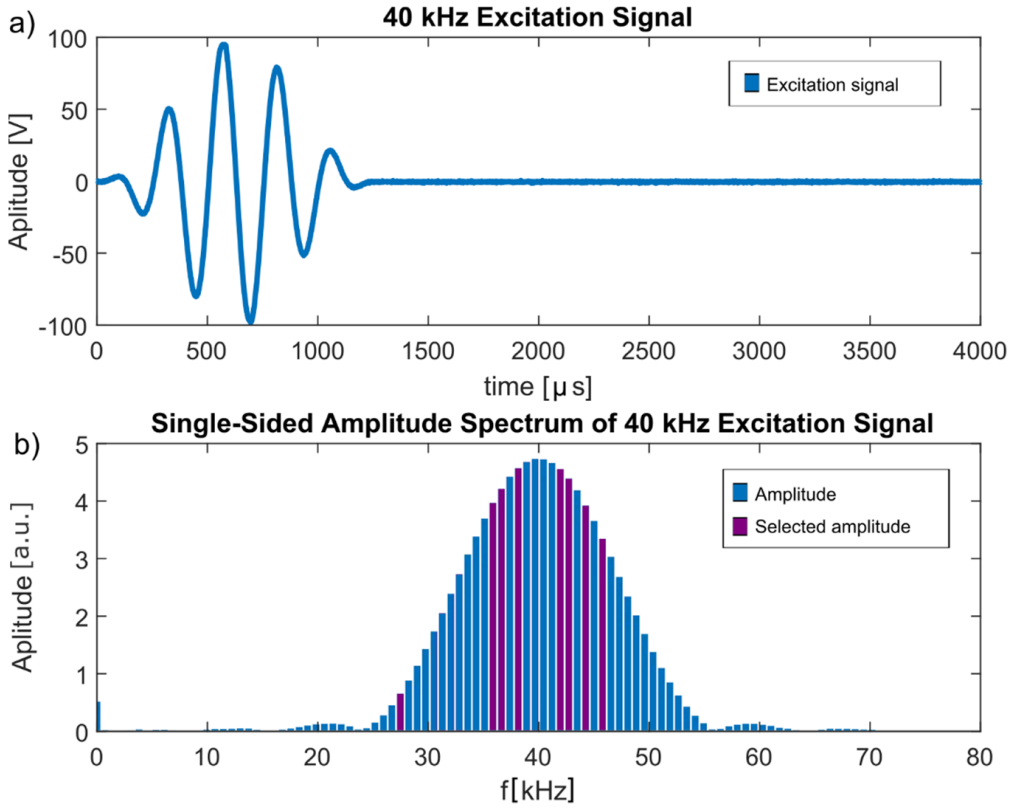


Figure 7. (**a**) 40 kHz excitation signal. (**b**) Single-sided amplitude spectrum of the 40 kHz excitation signal. Purple bars indicate the frequencies selected by the improved algorithm of the automated toolbox.

3.3. Influence of the Distance between Damage Location and Signal Path

Incorrectly classified data samples resulted mostly from signals of damage location D_{24}, which required a considerable extrapolation since this damage location was furthest from the signal path (186 mm; Table 1), which is believed to have had a significant influence on the ML performance, especially at higher temperatures. Therefore, we performed an additional investigation of the combination of transducers 1 and 7 (Table 7), where D_{24} lay in the direct signal path. Table 8 shows the distances of each damage location from the direct signal path for this transducer combination.

Table 7. Position of transducers 1 and 7.

Label	Position on x-Axis (mm)	Position on y-Axis (mm)
Transducer 1	450	470
Transducer 7	450	30

Table 8. Distance of the damage locations from the signal path between transducers 1 and 7.

Label	Distance from Signal Path (mm)
Damage 04	385
Damage 12	255
Damage 16	11.5
Damage 24	0

The results given in Table 9 show the same tendency as for the combination of transducers 4 and 9: D_{24} and D_{16} were close to the signal path; thus, they were classified correctly, whereas the accuracy dropped with increasing distance between damage location and signal path. The reduced accuracies for the undamaged cases (UG_1, UG_2) were possibly due to features present in the damage cases being similar to features of the undamaged case; however, this needs to be investigated further.

Table 9. Accuracy and number of misclassifications of the improved algorithm (BFC for feature extraction, RELIEFF for feature selection, SVM with RBF kernel for classification validated with LOGOCV) for the combination of transducers 1 (sender) and 7 (receiver).

Validation Results of the Improved Algorithm for the Combination of Transducers 1 and 7							
Damage case	UG_1	UG_2	D_{04}	D_{12}	D_{16}	D_{24}	Total
Misclassifications	4	39	133	68	0	0	244
Accuracy	94.9%	75.8%	17.4%	57.7%	100%	100%	72%

3.4. Robustness against Temperature Influences

The temperature range tested by Moll et al. [12] simulates conditions from room temperature up to 60 °C in 0.5 °C steps, making it suitable primarily for indoor applications, e.g., lightweight manipulators for robots [37]. To also cover outdoor applications, e.g., rotor blades of wind turbines, which have to withstand temperatures in the range from −50 °C to +100 °C [38], the temperature range needs to be extended in future experiments. To investigate the influence of a smaller temperature range while training the ML model, i.e., to check how well the model can extrapolate, a training temperature range was successively reduced, extending the required extrapolation from 2 °C to 16 °C in 2 °C steps. In the scope of this manuscript, extrapolation denotes testing of measurements that were performed outside the trained temperature range. Thus, a model was first built using the temperature range 22.5 °C to 57.5 °C for training and validation, then it was tested for the temperature ranges 20 °C to 22 °C and 58 °C to 60 °C, and then further the training range was further reduced and the test temperature range increased. Within each case, data from UG_1, D_{12}, and D_{24} were used for training, and data from D_{04} and the rising temperature flank of UG_2 for validation. The extended temperature range of these data plus the respective data from D_{16} and the descending flank of UG_2 were used for testing, as shown in Figure 8a,b for 2 °C and 16 °C extrapolation, respectively.

Note that further extrapolation is not meaningful since the size of the training data set was reduced with every step, decreasing the statistical significance. For 16 °C extrapolation, the training data (green areas in Figure 8b) only contained 75 measurements in the range of 36.5 °C to 43.5 °C.

Table 10 shows the test accuracies achieved for each temperature extrapolation step. The ML model extrapolated up to 6 °C without loss of performance and had only a slight decrease in performance for temperature extrapolations up to 10 °C, indicating that the model is fairly robust to temperature influences. This might allow a model to be built based on data from a lab environment that could still achieve acceptable performance under real operating conditions. Note that extrapolation over 12 °C corresponds to a training range from 32.5 °C to 47.5 °C, i.e., ΔT = 15 °C. Thus, only approx. one third of the overall temperature range is necessary to achieve an accuracy of 93.6% even for previously unknown damage locations.

Table 10. Resulting testing accuracy over temperature extrapolation. The extrapolated temperatures were not used for the model building and only used for testing.

Resulting Testing Accuracy for a Certain Temperature Extrapolation							
Temperature extrapolation	2 °C	4 °C	6 °C	8 °C	10 °C	12 °C	14 °C
Testing accuracy	100%	100%	100%	97.0%	96.8%	93.6%	83.7%

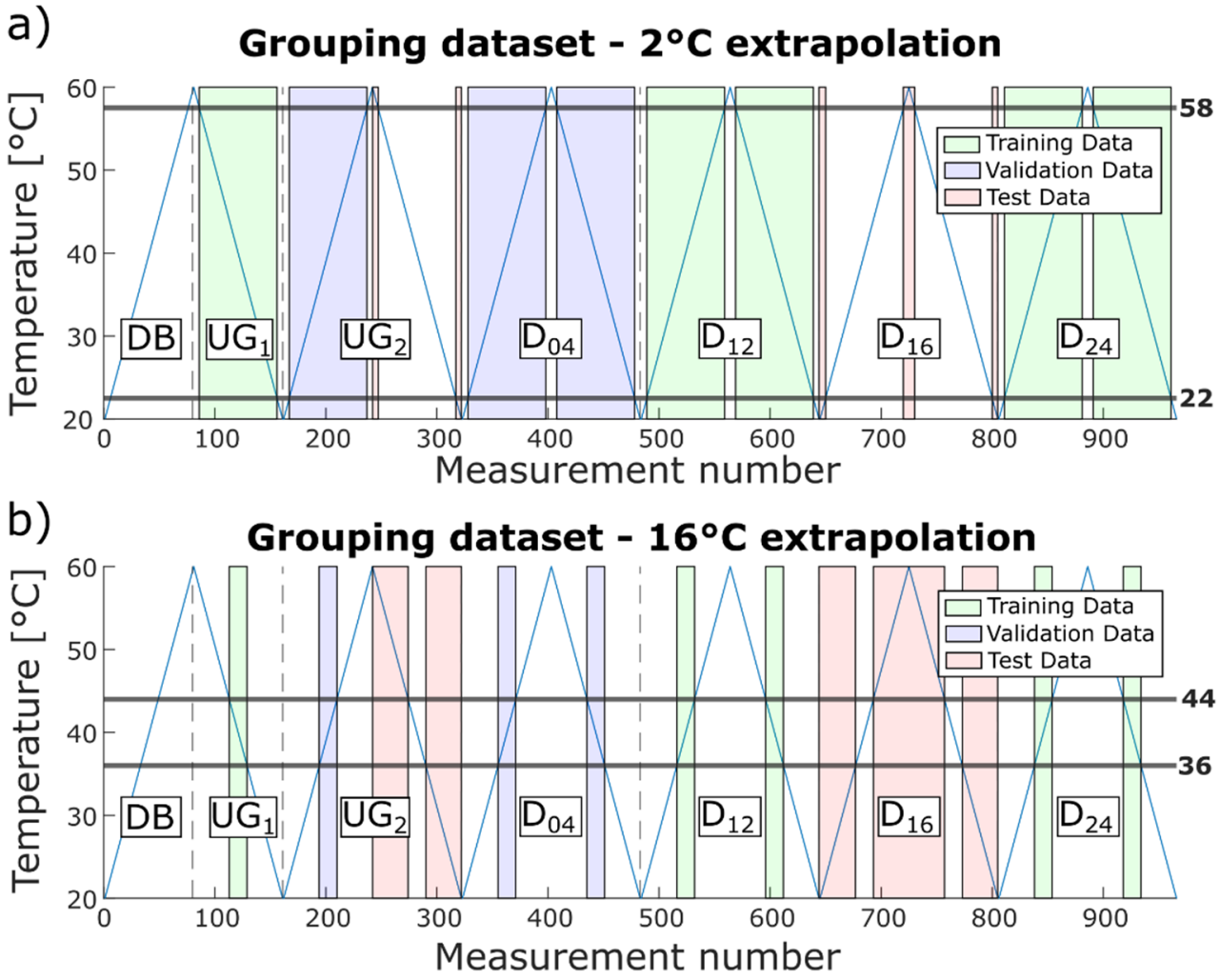


Figure 8. Grouping of the data into training, validation, and test data for (**a**) 2 °C and (**b**) 16 °C extrapolation.

3.5. Comparison to a State-of-the-Art Neural Network

Since neural networks (NN) are nowadays often used for SHM applications [39–41], we benchmarked our approach against a neural network approach reported for the same dataset [9]. In this study, Mariani et al. first tested several deep learning algorithms, namely, a multilayer perceptron, a recurrent neural network with long short-term memory, and a WaveNet-based causal dilated convolutional neural network (CNN), on a reference guided wave SHM dataset using a threshold-based OBS + BSS as the benchmark. They found that multilayer perceptrons and recurrent neural networks were not able to significantly outperform OBS + BSS, whereas the causal dilated CNN delivered high accuracy within reasonable training time and was therefore applied to the experimental guided wave dataset for varying temperature [12]. Mariani et al. achieved 100% accuracy on the testing data for the transducer combination T_4 to T_{10} with a high-pass filter (Butterworth), down sampling (factor 6), and BSS (undamaged plate at 40 °C) as pre-processing. A more detailed description as well as the architecture of the causal dilated CNN can be found in the original paper [9].

To compare our approach with these results for the causal dilated CNN, we also evaluated the transducer combination T_4 and T_{10} for model building and replicated the grouping of Mariani et al. for training, validation, and testing data. Thus, training data contained D_{16}, D_{24}, and 50% of UG_2; validation data contained D_{12} and 25% of UG_2; and testing data contain D_{04} and 25% of UG_2. The split of UG_2 into the corresponding groups was based on a training–validation–training–testing pattern with a 1.5 °C step size (e.g.,

data from 20 °C–21.5 °C were used for training, 22 °C–23.5 °C for validation, 24 °C–25.5 °C for training, 26 °C–27.5 °C for testing, 28 °C–29.5 °C again for training, etc.).

The model was built using the improved approach described above, with BFC as a feature extractor, PCC for feature pre-selection, RELIEFF for the final feature selection, and SVM with RBF kernel as a classifier. Out of the possible combinations for the hyper-parameters, the algorithm selected 30 as the best number of features and 10,000 as the value for parameter C. Actually, a wide range of hyper-parameter combinations achieved a validation accuracy of 100%, showing that the approach is robust (Appendix C, Figure A2). After hyper-parameter selection and before applying the model on the test data, it was again trained with all training and validation data. The achieved prediction accuracy of 100% for damage D_{04} matches the result reported by Mariani et al.

The computational time for our model was 185 s on an Intel® Core™ i7 8650U CPU, which is also similar to the 5 min training time for the causal dilated CNN reported by Mariani et al. using one NVIDIA® Quadro RTX™ 6000 GPU (2000 epochs). Note, however, that the CPU used in our study only has a theoretical computational performance of 0.442 TFLOPS (tera floating-point operations per second) compared to 16.3 TFLOPS of the GPU.

At first glance it might seem that the causal dilated CNN required less data pre-processing. However, hyper-parameter optimisation (HPO) is not described by Mariani et al. in their study. It is well known that HPO of NN models often requires significant (hardware and human) resources. Over the last few years, different approaches [42–44] have been proposed to solve this problem. Existing methods and frameworks to find a proper architecture and HPO of NNs are often computationally expensive and/or application-specific [43,44]. On the other hand, HPO for our proposed approach is simple and clear, as demonstrated by Figure A2 (Appendix C), which is one of the advantages of using classical ML methods (feature extraction/feature selection/simple classification) instead of deep NN models. Furthermore, our approach directly provides relevant features, i.e., a physically interpretable result, whereas NN models are often a black box and require significant additional effort to allow for interpretation.

4. Conclusions

This paper presents results of an automated ML framework applied to damage detection for guided wave-based structural health monitoring. We demonstrate that damage locations were correctly classified with a success rate of 88% without domain-specific knowledge or hyper-parameter tuning. By interpreting the results of the automated toolbox and a slight tuning of the hyper-parameters, an accuracy of 96.2% was achieved using a realistic group-based validation scenario while keeping the improvement time and effort low and, more importantly, achieving physically interpretable results.

Due to the small dataset size (for a single transducer combination T4 to T10 at 40 kHz excitation frequency) with the unbalanced ratio between the number of measurements for damaged and undamaged structures, plus the lab setup with reduced ambient influences, no conclusion can be drawn regarding how well the approach would perform in real-life applications. Edge reflections, boundary conditions, and complex geometries might lead to lower performance.

Therefore, application of the presented ML framework on real damages and CFRP components in extended temperature ranges (e.g., −50 °C to +100 °C), as well as the influence of the distance between sensors and damages, edge effects, and other damage types, offer an interesting field for future research.

Supplementary Materials: The Automated ML Toolbox for Cyclic Sensor Data can be downloaded at: https://github.com/ZeMA-gGmbH/LMT-ML-Toolbox (accessed on 28 December 2021); The Automated ML Toolbox DAV^{3}E can be downloaded at: https://www.lmt.uni-saarland.de/index.php/de/forschung/157-dav3e (accessed on 28 December 2021).

Author Contributions: Conceptualisation, C.S., T.S., J.M., K.T. and Y.L.; methodology, C.S., J.B., K.T. and J.M.; software, T.S. and C.S.; validation, C.S., P.G. and T.S.; formal analysis, C.S.; investigation, J.M. and J.P.; resources, J.M. and A.S.; data curation, J.M. and T.S.; writing—original draft preparation, C.S.; writing—review and editing, T.S., A.S., J.M., K.T., Y.L., J.P. and J.B.; visualisation, C.S.; supervision, T.S.; project administration, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: The APC was funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) and Saarland University within the funding programme Open Access Publishing.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data "Guided wave data for varying temperature" presented in this study are openly available in the Open Guided Waves Platform at https://doi.org/10.6084/m9.figshare.9863465 [12].

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ALA	Adaptive linear approximation
BDW	Best Daubechies wavelets
BFC	Best Fourier coefficients
BSS	Baseline signal stretch
CFRP	Carbon fibre-reinforced plastic
CNN	Convolutional neural network
CPU	Central processing unit
CV	Cross-validation
D_{XX}	Damage number XX
DB	Database
FE	Feature extraction
FS	Feature selection
GPU	Graphics-processing unit
GW	Guided waves
HPO	Hyper-parameter optimisation
LDA	Linear discriminant analysis
LOGOCV	Leave-one-group-out cross-validation
ML	Machine learning
NN	Neural network
OBS	Optimal baseline selection
PC	Principle component
PCA	Principle component analysis
PCC	Pearson correlation coefficient
RBF-Kernel	Radial basis function kernel
RFE-SVM	Recursive feature elimination support vector machines*
RH	Relative humidity
RMSE	Root mean square error
SHM	Structural health monitoring
SM	Statistical moments
SVM	Support vector machines
T_X	Transducer number X
TFLOPS	Tera floating-point operations per second
UG	Undamaged group

Appendix A

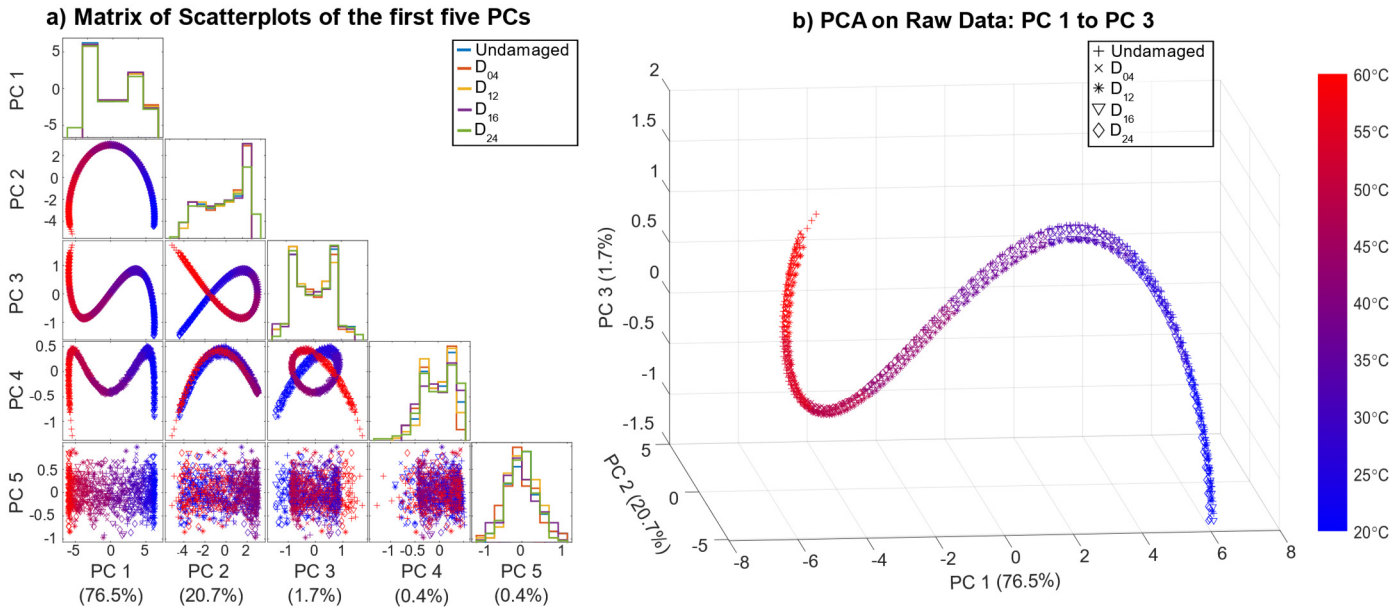


Figure A1. (**a**) Matrix of the first five PCs of the PCA on the raw data (undamaged plate and all simulated damage locations) coloured by their corresponding temperature with their histograms on the diagonal and the variance explained by each PC given as percentage in brackets. (**b**) First three PCs plotted into a three-dimensional space.

Appendix B

The following section describes the mathematical background of the applied ML algorithms (BFC, PCC, RELIEFF, RFE-SVM).

First, from the pre-processed signal with 13,108 samples per measurement, the frequency domain representation is calculated by using a discrete Fourier transform (A1) as well as the corresponding phase angles. Here, the standard implementations *fft()* and *phase()* of MATLAB 2021a are used [45,46]. It holds

$$Y(k) = \sum_{j=1}^{n} X(j) W_n^{(j-1)(k-1)} \quad with\ W_n = e^{-2\pi i/n}, \tag{A1}$$

where $Y(k)$ denotes the Fourier transform of the input signal X with length n and the imaginary unit i.

The resulting two-sided spectrum is converted into a single-sided amplitude spectrum. All necessary steps can be found in [45]. Next, the computed frequencies are ranked according to the absolute value of their amplitudes, and the highest 10% (1310 amplitudes) with their corresponding phase angle (1310 angles) are used as features (2620 features).

For the first feature (pre-)selection step with Pearson linear correlation coefficient r down to 500 features, it holds

$$\begin{aligned} r(a,b) &= \frac{\sum_{i=1}^{n}\left(X_{a,i}-\overline{X}_a\right)\left(Y_{b,i}-\overline{Y}_b\right)}{\sqrt{\left\{\sum_{i=1}^{n}\left(X_{a,i}-\overline{X}_a\right)^2\left(Y_{b,i}-\overline{Y}_b\right)^2\right\}}} \\ &\text{with } \overline{X_a} = \tfrac{1}{n}\sum_{i=1}^{n}\left(X_{a,i}\right), \\ &\text{and } \overline{Y_b} = \tfrac{1}{n}\sum_{j=1}^{n}\left(X_{b,j}\right), \end{aligned} \tag{A2}$$

where X denotes the matrix of pre-selected features and Y the target. $X_a \in \mathbb{R}^{n\times 1}$ represents a column of matrix X and $Y_b \in \mathbb{R}^{n\times 1}$ a column of matrix Y.

Before applying RELIEFF as main feature selection method, the preselected features get standardised. RELIEFF is implemented in MATLAB by using the built-in *knnsearch()* function to determine the indexes of the three nearest neighbours (city block distance metric) of the same group (hits), and the nearest neighbours of the other groups (misses) [47]. The features are eventually ranked, with the features with a high distance to other groups (misses) and low distance to the same group (hits) achieving a higher ranking. Another internal 10-fold CV determines the necessary number of selected features.

The classifier support vector machine with radial basis function kernel (RBF kernel) tries to find a multidimensional hyperplane

$$\vec{w}, \vec{x} + b = 0\,, \tag{A3}$$

with $\vec{w}$ being a normal vector and b the bias term to optimally separate two classes [32]. The goal of training an L1-norm SVM is to maximise the generalisability of the model towards untrained data by minimising

$$Q\left(\vec{w}, b, \vec{\xi}\right) = \frac{1}{2}\left|\vec{w}\right|^2 + C\sum_{i=1}^{M}\xi_i, \tag{A4}$$

as shown in [31].

Misclassifications need to be tolerated but kept track of using the parameter $\vec{\xi}$, where C acts as a regularisation parameter. Depending on which side of this hyperplane new datapoints appear on, they are classified as either class one or class two. To also separate data that show non-linear behaviour, the so-called kernel trick transforms the data into a higher dimensional feature space, in which the hyperplane might be able to linearly separate the two classes. The chosen RBF kernel (5) transforms data into an infinite-dimensional feature space. Here, every support vector is the centre point of a radial Gaussian function

$$K\left(\vec{x}, \vec{x}'\right) = \exp\left(-\frac{\left|\vec{x} - \vec{x}'\right|}{2\sigma}\right) \tag{A5}$$

where σ corresponds to the radius of the Gaussian function. Note that the parameter σ is automatically optimised in an heuristic procedure by the MATLAB function *fitcecoc()* [48] while using *templateSVM()* [49] with *KernelScale* set to *auto*. To ensure reproducibility, a seed (*default*, respectively 0) is specified for the random number generator of MATLAB. This results in the following optimization problem [31,32]:

$$\text{maximise } Q(\alpha) = \sum_{i=1}^{M}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{M}\alpha_i\alpha_j y_i y_j K\left(\vec{x}_i, \vec{x}_j\right)\,, \tag{A6}$$

where M denotes the number of α non-negative Lagrange Multiplicators, y the class, and $K\left(\vec{x}_i, \vec{x}_j\right)$ the kernel function. Once the SVM is trained, new data can be classified by using

$$D\left(\vec{x}\right) = \sum_{i\in S}\alpha_i y_i K\left(\vec{x_i}, \vec{x_j}\right) + b \text{ is classified into } \begin{cases} \text{Class 1, } if\ D\left(\vec{x}\right) > 0 \\ \text{Class 2, } if\ D\left(\vec{x}\right) < 0 \end{cases}, \tag{A7}$$

where S denotes the set of support vector indices. Strategies for handling multiclass classification problems can be found in [31].

Appendix C

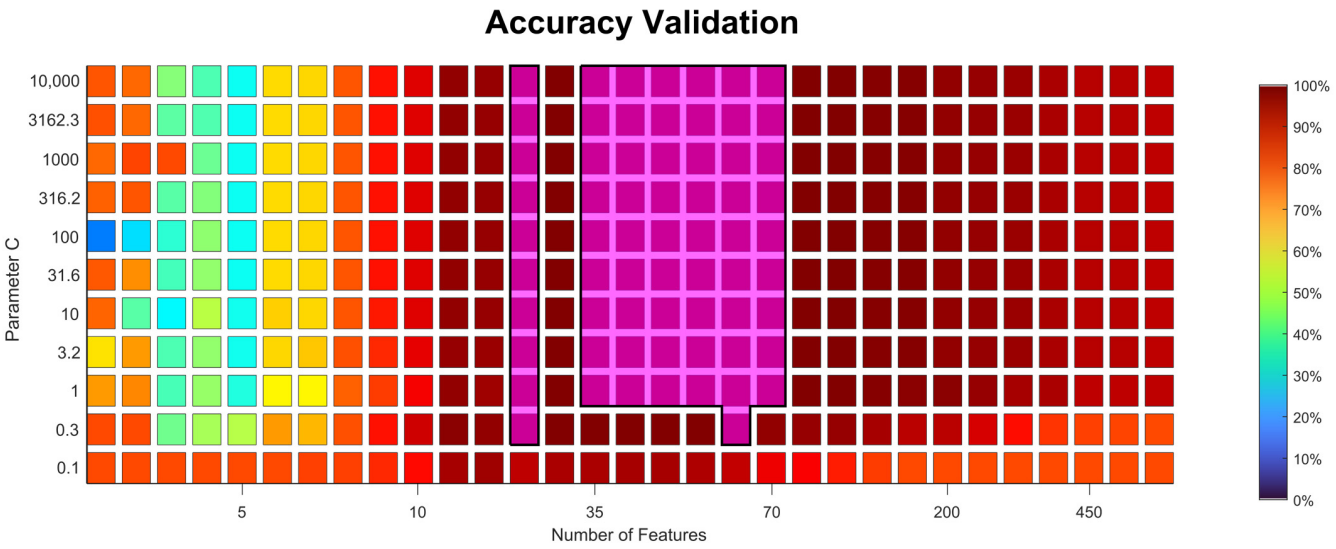


Figure A2. Resulting accuracy of the validation data for various parameter combinations (parameter *C* of the SVM and number of features selected by RELIEFF). Parameter combinations within the purple boxes achieve 100% validation accuracy. The gap between the two purple boxes consists of parameter combinations achieving an accuracy of 99.5%.

References

1. Qiu, L.; Fang, F.; Yuan, S. Improved density peak clustering-based adaptive Gaussian mixture model for damage monitoring in aircraft structures under time-varying conditions. *Mech. Syst. Signal Process.* **2019**, *126*, 281–304. [CrossRef]
2. Abdeljaber, O.; Sassi, S.; Avci, O.; Kiranyaz, S.; Ibrahim, A.A.; Gabbouj, M. Fault Detection and Severity Identification of Ball Bearings by Online Condition Monitoring. *IEEE Trans. Ind. Electron.* **2018**, *66*, 8136–8147. [CrossRef]
3. Munir, N.; Kim, H.-J.; Park, J.; Song, S.-J.; Kang, S.-S. Convolutional neural network for ultrasonic weldment flaw classification in noisy conditions. *Ultrasonics* **2018**, *94*, 74–81. [CrossRef]
4. De Oliveira, M.A.; Monteiro, A.V.; Filho, J.V. A New Structural Health Monitoring Strategy Based on PZT Sensors and Convolutional Neural Network. *Sensors* **2018**, *18*, 2955. [CrossRef]
5. Cruz, F.; Filho, E.S.; de Albuquerque, M.C.S.; Silva, I.; Farias, C.; Gouvêa, L. Efficient feature selection for neural network based detection of flaws in steel welded joints using ultrasound testing. *Ultrasonics* **2016**, *73*, 1–8. [CrossRef]
6. Bornn, L.; Farrar, C.R.; Park, G.; Farinholt, K. Structural Health Monitoring With Autoregressive Support Vector Machines. *J. Vib. Acoust.* **2009**, *131*, 021004. [CrossRef]
7. Roy, S.; Chang, F.; Lee, S.; Pollock, P.; Janapati, V. A novel machine-learning approach for structural state identification using ultrasonic guided waves. In *Safety, Reliability, Risk and Life-Cycle Performance of Structures and Infrastructures*; Deodatis, G., Ellingwood, B., Frangopol, D., Eds.; CRC Press: Boca Raton, FL, USA, 2014; pp. 321–328.
8. Miorelli, R.; Kulakovskyi, A.; Chapuis, B.; D'Almeida, O.; Mesnil, O. Supervised learning strategy for classification and regression tasks applied to aeronautical structural health monitoring problems. *Ultrasonics* **2021**, *113*, 106372. [CrossRef]
9. Mariani, S.; Rendu, Q.; Urbani, M.; Sbarufatti, C. Causal dilated convolutional neural networks for automatic inspection of ultrasonic signals in non-destructive evaluation and structural health monitoring. *Mech. Syst. Signal Process.* **2021**, *157*, 107748. [CrossRef]
10. Keogh, E.; Kasetty, S. On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration. *Data Min. Knowl. Discov.* **2003**, *7*, 349–371. [CrossRef]
11. Schneider, T.; Helwig, N.; Schutze, A. Automatic feature extraction and selection for condition monitoring and related datasets. In Proceedings of the 2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Houston, TX, USA, 14–17 May 2018; pp. 1–6.
12. Moll, J.; Kexel, C.; Pötzsch, S.; Rennoch, M.; Herrmann, A.S. Temperature affected guided wave propagation in a composite plate complementing the Open Guided Waves Platform. *Sci. Data* **2019**, *6*, 1–9. [CrossRef]
13. Schubert, K.; Stieglitz, A.; Christ, M.; Herrmann, A. Analytical and Experimental Investigation of Environmental Influences on Lamb Wave Propagation and Damping Measured with a Piezo-Based System. *Work. Struct. Health Monit.* **2012**, *49*, 1–9.
14. Croxford, A.J.; Moll, J.; Wilcox, P.; Michaels, J.E. Efficient temperature compensation strategies for guided wave structural health monitoring. *Ultrasonics* **2010**, *50*, 517–528. [CrossRef]
15. Schneider, T.; Helwig, N.; Schütze, A. Industrial Condition Monitoring with Smart Sensors Using Automated Feature Extraction and Selection. *Meas. Sci. Technol.* **2018**, *29*, 094002. [CrossRef]

16. Dorst, T.; Robin, Y.; Schneider, T.; Schütze, A. Automated ML Toolbox for Cyclic Sensor Data. In Proceedings of the Mathematical and Statistical Methods for Metrology MSMM, Virtual, 31 May–1 June 2021.
17. Bastuck, M.; Baur, T.; Schütze, A. DAV3E—A MATLAB Toolbox for Multivariate Sensor Data Evaluation. *J. Sens. Sens. Syst.* **2018**, *7*, 489–506. [CrossRef]
18. Olszewski, R. *Generalized Feature Extraction for Structural Pattern Recognition in Time-Series Data*; Carnegie Mellon University: Pittsburgh, PA, USA, 2001.
19. Wold, S.; Esbensen, K.; Geladi, P. Principal Component Analysis. *Chemom. Intell. Lab. Syst.* **1987**, *2*, 37–52. [CrossRef]
20. Mörchen, F. *Time Series Feature Extraction for Data Mining Using DWT and DFT*; University of Marburg: Marburg, Germany, 2003.
21. Daubechies, I. *Ten Lectures on Wavelets*; SIAM: Philadelphia, PA, USA, 1992.
22. Papoulis, A.; Pillai, S.U. *Probability, Random Variables, and Stochastic Processes*; McGraw-Hill: Boston, MA, USA, 2002.
23. Guyon, I.; Elisseeff, A. An Introduction to Variable and Feature Selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182.
24. Rakotomamonjy, A. Variable Selection Using SVM-Based Criteria. *J. Mach. Learn. Res.* **2003**, *3*, 1357–1370.
25. Robnik-Šikonja, M.; Kononenko, I. Theoretical and Empirical Analysis of ReliefF and RReliefF. *Mach. Learn.* **2003**, *53*, 23–69. [CrossRef]
26. Kononenko, I.; Šimec, E.; Robnik-Šikonja, M. Overcoming the Myopia of Inductive Learning Algorithms with RELIEFF. *Appl. Intell.* **1997**, *7*, 39–55. [CrossRef]
27. Cohen, I.; Huang, Y.; Chen, J.; Benesty, J.; Chen, Y.H.; Cohen, I. Pearson Correlation Coefficient. In *Noise Reduction in Speech Processing*; Springer: Berlin/Heidelberg, Germany, 2009. [CrossRef]
28. Duda, R.O.; Hart, P.E.; Stork, D.G. *Pattern Classification*; Wiley: New York, NY, USA, 2001.
29. Wainer, J. Comparison of 14 Different Families of Classification Algorithms on 115 Binary Datasets. *arXiv* **2016**, arXiv:1606.00930.
30. Gui, G.; Pan, H.; Lin, Z.; Li, Y.; Yuan, Z. Data-driven support vector machine with optimization techniques for structural health monitoring and damage detection. *KSCE J. Civ. Eng.* **2017**, *21*, 523–534. [CrossRef]
31. Abe, S. *Support Vector Machines for Pattern Classification*; Springer: London, UK; New York, NY, USA, 2010.
32. Schölkopf Bernhard, B. *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*; MIT Press: Cambridge, MA, USA, 2018.
33. Liu, S.; Jiang, N. SVM Parameters Optimization Algorithm and Its Application. In Proceedings of the 2008 IEEE International Conference on Mechatronics and Automation, Takamatsu, Japan, 5–8 August 2008; pp. 509–513.
34. Liu, Y.; Du, J. Parameter Optimization of the SVM for Big Data. In Proceedings of the 2015 8th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 12–13 December 2015; pp. 341–344.
35. Hsu, C.; Chang, C.; Lin, C.-J. *A Practical Guide to Support Vector Classification*; University of National Taiwan: Taipei, Taiwan, 2003.
36. Schnur, C.; Moll, J.; Lugovtsova, Y.; Schütze, A.; Schneider, T. Explainable Machine Learning For Damage Detection: In Carbon Fiber Composite Plates Under Varying Temperature Conditions. In Proceedings of the ASME 2021 48th Annual Review of Progress in Quantitative Nondestructive Evaluation, Virtual, 28–30 July 2021.
37. Kartikejan, P.; Sabarianand, D.; Sugantan, S. Investigation on adaptability of carbon fiber tube for serial manipulator. *FME Trans.* **2019**, *47*, 412–417. [CrossRef]
38. Skolaut, W. (Ed.) *Maschinenbau: Ein Lehrbuch für das Ganze Bachelor-Studium*; Springer: Berlin, Germany, 2014.
39. Liu, D.; Tang, Z.; Bao, Y.; Li, H. Machine-Learning-Based Methods for Output Only Structural Modal Identification. *Struct. Control. Health Monit.* **2021**, *28*, e2843. [CrossRef]
40. XXu, L.; Yuan, S.; Chen, J.; Ren, Y. Guided Wave-Convolutional Neural Network Based Fatigue Crack Diagnosis of Aircraft Structures. *Sensors* **2019**, *19*, 3567. [CrossRef]
41. Azimi, M.; Eslamlou, A.D.; Pekcan, G. Data-Driven Structural Health Monitoring and Damage Detection through Deep Learning: State-of-the-Art Review. *Sensors* **2020**, *20*, 2778. [CrossRef]
42. Xu, Y.; Xie, L.; Zhang, X.; Chen, X.; Qi, G.-J.; Tian, Q.; Xiong, H. PC-DARTS: Partial Channel Connections for Memory-Efficient Architecture Search. *arXiv* **2020**, arXiv:1907.05737.
43. Zoph, B.; Le, Q.V. Neural Architecture Search with Reinforcement Learning. *arXiv* **2017**, arXiv:1611.01578.
44. Liu, H.; Simonyan, K.; Yang, Y. DARTS: Differentiable Architecture Search. *arXiv* **2019**, arXiv:1806.09055.
45. Fast Fourier Transform—MATLAB Fft—MathWorks Deutschland. Available online: https://de.mathworks.com/help/matlab/ref/fft.html#buuutyt-6 (accessed on 20 July 2021).
46. Phase Angle—MATLAB Angle—MathWorks Deutschland. Available online: https://de.mathworks.com/help/matlab/ref/angle.html (accessed on 20 July 2021).
47. Find K-Nearest Neighbors Using Input Data—MATLAB Knnsearch—MathWorks Deutschland. Available online: https://de.mathworks.com/help/stats/knnsearch.html (accessed on 20 July 2021).
48. Fit Multiclass Models for Support Vector Machines or Other Classifiers—MATLAB Fitcecoc—MathWorks Deutschland. Available online: https://de.mathworks.com/help/stats/fitcecoc.html#bufm0tv (accessed on 9 December 2021).
49. Support Vector Machine Template—MATLAB TemplateSVM—MathWorks Deutschland. Available online: https://de.mathworks.com/help/stats/templatesvm.html (accessed on 9 December 2021).

6.4 Further Spread of the Toolbox

To make the toolbox commonly accessible, it was published under an open-source license on GitHub [127]. Additionally, it was integrated into two different frameworks (DAV³E and Odion Digital Factory), as described in the following sections.

6.4.1 EaSy-ML

To enable the utilization of the automated machine learning Toolbox in an industrial environment, five use cases based on the toolbox were integrated into the ODION Digital Factory [128] product throughout the project EaSy-ML [129]. They allow the user to analyze data recorded by the ODION Digital Factory in the following use cases:

- Visualization of correlations between parameters
- Novelty and anomaly detection
- Sensor error detection
- Product quality monitoring
- Process quality monitoring

In each use-case, the user is supported by a digital assistant, as much automation as possible, and numerous visualizations so that only a short introduction to the software instead of extensive machine learning knowledge is required. Simultaneously, the graphical user interface removes the necessity to write any code and enables maintenance staff to use the program.

6.4.2 DAV³E

DAV³E [130] is short for Data Analysis and Verification/Visualization/Validation Environment. It constitutes a MATLAB toolbox for feature extraction from cyclic sensor signals, sensor fusion, data preprocessing, and statistical model building and evaluation. DAV³E provides no-code AI through a fully graphical user interface. The individual algorithms from the automated machine learning toolbox were added to DAV³E as well as additional options for manual hyperparameter tuning. DAV³E has been published under open source license on GitHub [131].

6.4.3 Personal Information Assistant

The Personal Information Assistant (PIA) [132] is a web-based front end developed by Schnur to facilitate and document the usage of a checklist for measurement and data planning for machine learning projects [121]. PIA comprises three modules: accessibility of (meta)data and knowledge, measurement and data planning, and data analysis.

6.4.4 Ongoing Extensions and Further Research Inspired by the Automated Machine Learning Toolbox

One drawback of the algorithmic composition described in Papers 1-3 is that one feature extraction algorithm is chosen to be applied to all sensors. However, a dataset could contain sensors with information best extracted by different extractors. Also, using brute force to select both algorithms and the number of selected features prohibits using more advanced non-linear classification or regression algorithms. Both problems were addressed by Pültz [85], who combined the automated machine learning toolbox with Bayesian Optimization [133], which is used to search for the best feature extraction algorithm for each sensor and simultaneously the best feature number to select. This replacement of exhaustive search with Bayesian optimization enables choosing a better feature extractor and using other classifiers and regressors like Support Vector Machines by greatly reducing the number of calls to the classifier/regressor training function.

Also, extensive comparisons with multiple common neural network architectures and using neural networks for classification, regression, and feature selection have been performed and are in preparation for publication [56].

7 Discussion and Future Work

In conclusion, this thesis answered the question of how to utilize automatic machine learning in condition monitoring and how to address all associated issues at once by designing an automated machine learning toolbox of complementary algorithms. More specifically, Paper 1 answered which algorithms and hyperparameters to choose by introducing the toolbox. Paper 2 answered how to apply those algorithms by demonstrating properties like explainability in multiple examples. Moreover, Paper 3 answered the question of how to utilize novelty detection by developing detailed instructions for three different application goals: outlier detection, monitoring of supervised learning, and detection of previously unknown faults.

As stated in Section 1, any solution to automatic machine learning in industrial condition monitoring must address multiple issues simultaneously. As demonstrated in numerous application scenarios in this thesis, no other approach does it better. For the toolbox developed in this thesis, the issues are addressed as follows:

The vast amount of data encountered in industrial condition monitoring refers to the massive amount of data recorded for each sample. It mainly affects feature extraction and partially affects feature selection. Therefore, all feature extraction algorithms employed in the toolbox are highly scalable in the amount of data per sample, with the worst being the FFT in BFC that, in general, is still considered to be computationally cheap. Additionally, univariate feature pre-selection using Pearson correlation is linearly scalable in the number of samples and the number of features per sample. Note that univariate pre-selection neglects feature interactions, possibly affecting model performance. During Training, feature extraction and pre-selection support parallelization through Map-Reduce, which has been demonstrated for multiple TB of lifetime measurements of electromechanical cylinders.

The **limited bandwidth to the cloud** is effectively targeted by the toolbox's ability to infer feature extraction and selection on edge devices, which limits the amount of data per sample to a maximum but usually well lower number of 500 features. This approach eliminates the need to communicate raw data, cutting the required bandwidth by multiple orders of magnitude. Note that by doing so, all additional goals beyond toolbox inference, like online learning, are then limited to the selected features which might impact performance.

Scalable algorithms solve the issue of **limited resources on the edge**, like the vast amount of data addressed. CANWAY Technologies GmbH and Fraunhofer IIS have

done such implementation in KI-Predict [30]. Additionally, the suggested feature extraction algorithms can be implemented using streaming algorithms, eliminating the need to cache the complete data of the sample to infer. This allows for the use of significantly cheaper edge devices.

The complete automation of algorithm and hyperparameter selection in supervised fault detection and detailed instructions for novelty detection addresses the **limited availability of machine learning experts**. This level of automation enables process engineers to apply the toolbox with very little training and little experience in machine learning, thus eliminating the need for dedicated machine learning experts. However, no automated approach to machine learning can guarantee optimal results. Paper B successfully demonstrates the toolbox's performance against other automatic approaches; however, it does not compare to models designed by machine learning experts that are expected to outperform the toolbox.

The **limited trust in black box models** is addressed by the global explainability of all models created by the toolbox, from feature extraction to classification. This enables the identification of relevant sensors, signal areas, or frequency bands, as demonstrated in Paper 2. The results on the electromagnetic cylinder have easily been shown to be physically meaningful, which is extremely hard or even impossible for other automatic machine learning approaches. Note that while identifying and interpreting relevant features highly increases trust in the models, the interpretation is qualitative, and quantitative analysis is still generally complex for humans due to the number of features used for decision-making.

The issue of **very diverse sensor signals** is addressed by benchmarking multiple complementary algorithms and automatic decisions for the most fitting for the specific application. Specifically, as described in Paper 1, feature extraction can extract information from the overall signal shape (PCA), time domain (ALA), time-frequency domain (BFC), frequency domain (BFC), and statistical properties of the signal covering a wide variety of sensors typically employed for condition monitoring. Additionally, in feature selection, RFESVM offers high performance on classification problems that allow linear separability of classes. Complementarily, RELIEFF provides high performance to circular separability, and Pearson correlation offers fast means of pre-selection. This results in the high versatility of the toolbox, which has been demonstrated by high performance in multiple scenarios (Papers 1, 2, and B). These demonstrations only compare the toolbox's performance to other automatic approaches. Those results are not guaranteed to be optimal, and it is to be expected that algorithms specifically designed for the respective scenario will outperform the toolbox.

Finally, the **requirement for high robustness** is addressed by employing linear discriminant analysis in combination with Mahalanobis classification. In Paper B, This straightforward and robust classification algorithm is up to par with popular algorithm classes like neural networks in leave one group out cross-validation. Compared to other algorithms, it suffers less decline in prediction performance due to domain shift, which shows its robustness. However, domain shifts posing a transferability problem are still the largest source of error in a model learned by the toolbox as Paper D showed random variations in data to have only a minor influence on prediction quality.

Solving transferability problems will be critical for a future widespread application of machine learning-based condition monitoring, as resulting domain shifts are the most common problem encountered during the development of this thesis that led to ML models not achieving target accuracies. The underlying root cause is believed to be the limited number of training samples and especially their regularly encountered lack of statistical independence. Future research needs to solve or counteract those issues. As shown in Figure 7, learning from a low number of domains causes additional training issues like a weak correlation between validation and test error, which makes the induced performance degradation hard to predict. Also, as shown in Paper C, deviations expected from error propagation of sensor noise are negligible compared to the performance degradation observed under domain shift [100].

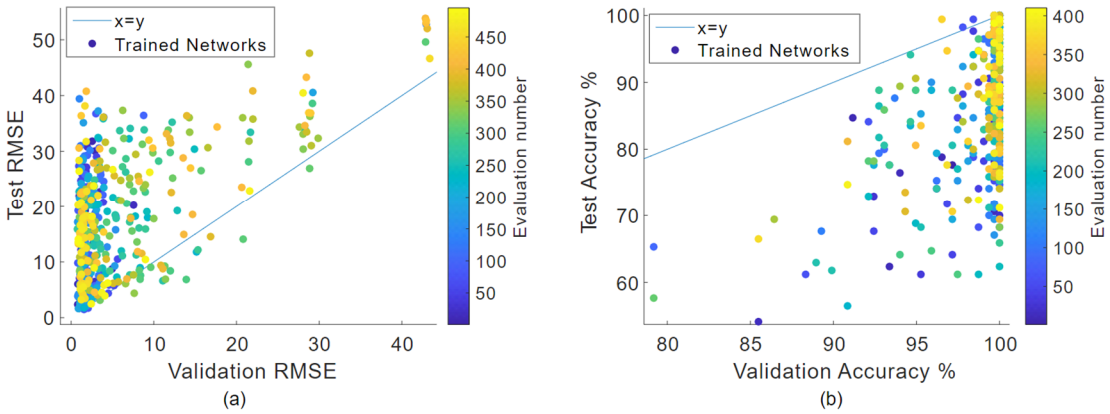


Figure 7: Results of a neural network architecture search under domain shift in test data. In this plot, the validation data are 20 % of the training set, which was randomly selected. The test data is from a different distribution, i.e., a different operating temperature. Each point is a trained network, (a) ZeMA hydraulic test bed detection of valve switching deterioration, (b) Case Western Reserver bearing dataset damage detection [100].

The following approaches can be investigated for their potential to solve the issue:

- Application of transfer-learning strategies: Domain Adaptation is a sub-category of transfer-learning commonly used in other application scenarios such as spam filtering. Multiple approaches to transferring a model from the source domain (training data) to a target domain (actual application data) have

been researched for those applications. Those can be categorized into four approaches. While reweighting algorithms and iterative algorithms rely on typically non-existing labeled data from the target domain, algorithms that aim for a common representation space of source and target domain [134, 135] or hierarchical Bayesian models seem promising candidates. Hierarchical Bayesian models construct a model allowing domain-specific and global latent variables [135]. Also, classical metrological approaches like calibration and adjustment can be seen as a transfer learning form compensating for an induced deviation. However, calibration requires labeled data from the target domain.

- Usage of extremely simple models: Using simple algorithms and strong regularization restricts the model to learning only the most basic correlation between patterns and targets. This is equivalent to treating the performance degradation induced by domain shifts as a classical overfitting effect. Although simple, this approach is only promising under the assumption of simple correlations between sensor patterns and targets that are also dominant enough to be significant throughout different domains. However, it is to be expected that this approach trades robustness against domain shifts for predictive performance.
- Manual feature extraction or selection: If the model trained on the source domain is physically interpretable, the gained insights can be used to create features invariant to expected domain shifts. Domain knowledge can compensate for observed domain shifts by suitable preprocessing or manually selecting physically meaningful features from those found to be statistically significant during feature selection. Therefore, such an approach would utilize physical knowledge to find domain shift-invariant features that would solve the domain shift problem. However, this approach is more work-intensive than those mentioned earlier. Additionally, qualitative model interpretations do not guarantee the quantitative interpretability needed to construct domain shift-invariant features.

The following approaches can be investigated or are already investigated by members of the DESS group at Saarland University and ZeMA for their potential to circumvent the issue:

- Identification of use cases with a high number of domains: The most straightforward way to circumvent domain shift problems is to find applications that offer a significant number of domain shifts in their training data. Such training data would implicitly allow learning algorithms to learn how to suppress or compensate for domain shifts. On the other hand, this

approach would ignore most condition monitoring use cases and severely limit applications to an extent where widespread use of ML-based condition monitoring is not expected. The guidelines created and developed by Christopher Schnur can help identify such use cases [121].

- In-production experiments for training data: Using data recorded during normal production for more training data can be another way for training algorithms to become robust against relevant domain shifts. Although training data experiments with a well-defined design of experiment are usually costly and commonly offer little training data, data from machines during mass production over a long period can be acquired at much lower costs. If this data could be partially labeled and recorded from different identical machines with sufficient production variety, the resulting database would allow an algorithm to become robust against domain shifts relevant during mass production. Steffen Klein investigates this approach in his thesis.
- Using model interpretation for physically motivated model building: This approach can be instrumental in optimization scenarios. The idea is to apply explainable AI algorithms to understand a model, which provides additional insight into the application and often helps provide physically motivated optimization for machines and processes. While unpublished due to high application security and data protection concerns, such scenarios were encountered multiple times during the work for this thesis. Typically, in these scenarios, ML was meant to circumvent an issue that resulted from physical effects or correlations missed by the process engineers and revealed by the statistical ML algorithm. That insight was used to solve the underlying issue instead of circumventing it.

Another issue specific to the automated machine learning toolbox revealed in more recent research is a blind spot concerning information in the signal envelope. Although signal envelope analysis is significant in all sorts of rotating machinery monitoring, this significance still needs to be reflected in the automated machine learning toolbox because the signal envelope was not relevant for one of the applications for which the toolbox was developed. By now, the issue has been fixed using the modular character, the easy expandability of the toolbox, and the inclusion of another feature extractor [127]. This again highlights the importance of the modular toolbox characteristics.

8 Sources:

[1] K. Schwab, *The Fourth Industrial Revolution*. Crown, 2016.

[2] H. Arnold, “Industrie 4.0: Ohne Sensorsysteme geht nichts,” Comment. Accessed: Mar. 10, 2018. [Online]. Available: https://www.elektroniknet.de/messen-testen/industrie-4-0-ohne-sensorsysteme-geht-nichts.110776.html

[3] D. Imkamp, J. Berthold, M. Heizmann, K. Kniel, E. Manske, M. Peterek, R. Schmitt, J. Seidler, and K. D. Sommer, “Challenges and trends in manufacturing measurement technology - The ‘Industrie 4.0’ concept,” *Journal of Sensors and Sensor Systems*, vol. 5, no. 2, pp. 325–335, Oct. 2016, doi: 10.5194/jsss-5-325-2016.

[4] K. D. Walter, “Wo bleibt der Sensor für Industrie 4.0.” Accessed: Mar. 10, 2018. [Online]. Available: http://www.elektrotechnik.vogel.de/ wo-bleibt-der-sensor-fuer-industrie-40-a-529141/

[5] Y. Duan, G. Fu, N. Zhou, X. Sun, N. C. Narendra, and B. Hu, “Everything as a Service (XaaS) on the Cloud: Origins, Current and Future Trends,” in *Proceedings - 2015 IEEE 8th International Conference on Cloud Computing, CLOUD 2015*, Institute of Electrical and Electronics Engineers Inc., Aug. 2015, pp. 621–628. doi: 10.1109/CLOUD.2015.88.

[6] M. Javaid, A. Haleem, R. P. Singh, and R. Suman, “Enabling flexible manufacturing system (FMS) through the applications of industry 4.0 technologies,” *Internet of Things and Cyber-Physical Systems*, vol. 2, pp. 49–62, Jan. 2022, doi: 10.1016/j.iotcps.2022.05.005.

[7] M. Javaid, A. Haleem, R. P. Singh, S. Rab, and R. Suman, “Significance of sensors for industry 4.0: Roles, capabilities, and applications,” *Sensors International*, vol. 2, p. 100110, Jan. 2021, doi: 10.1016/j.sintl.2021.100110.

[8] A. Varshney, N. Garg, K. S. Nagla, T. S. Nair, S. K. Jaiswal, S. Yadav, and D. K. Aswal, “Challenges in Sensors Technology for Industry 4.0 for Futuristic Metrological Applications,” *Mapan - Journal of Metrology Society of India*, vol. 36, no. 2, pp. 215–226, Jun. 2021, doi: 10.1007/s12647-021-00453-1.

[9] X. Li, Q. Ding, and J.-Q. Sun, “Remaining useful life estimation in prognostics using deep convolution neural networks,” *Reliability Engineering & System Safety*, vol. 172, pp. 1–11, Apr. 2018, doi: 10.1016/j.ress.2017.11.021.

[10] “iCM-Hydraulic – Data-based intelligent condition monitoring for hydraulic systems,” *Project*.

[11] A. Schütze, N. Helwig, and T. Schneider, “Sensors 4.0 - Smart sensors and measurement technology enable Industry 4.0,” *Journal of Sensors and Sensor Systems*, vol. 7, no. 1, pp. 359–371, Jan. 2018, doi: 10.5194/jsss-7-359-2018.

[12] NATIONAL INSTRUMENTS CORP., “CompactRIO-Systeme.” Accessed: Oct. 11, 2023. [Online]. Available: https://www.ni.com/de/shop/compactrio.html

[13] T. Stock and G. Seliger, “Opportunities of Sustainable Manufacturing in Industry 4.0,” *Procedia CIRP*, vol. 40, pp. 536–541, 2016, doi: 10.1016/j.procir.2016.01.129.

[14] V. Mathew, T. Toby, V. Singh, B. M. Rao, and M. G. Kumar, “Prediction of Remaining Useful Lifetime (RUL) of turbofan engine using machine learning,” in *2017 IEEE International Conference on Circuits and Systems (ICCS)*, IEEE, Dec. 2017, pp. 306–311. doi: 10.1109/ICCS1.2017.8326010.

[15] P. Franciosa, M. Sokolov, S. Sinha, T. Sun, and D. Ceglarek, “Deep learning enhanced digital twin for Closed-Loop In-Process quality improvement,” *CIRP Annals Manufacturing Technology*, vol. 69, no. 1, pp. 369–372, 2020, doi: 10.1016/j.cirp.2020.04.110.

[16] M. A. Zamora-Hernández, J. A. Castro-Vargas, J. Azorin-Lopez, and J. Garcia-Rodriguez, “Deep learning-based visual control assistant for assembly in Industry 4.0,” *Computers in Industry*, vol. 131, Oct. 2021, doi: 10.1016/j.compind.2021.103485.

[17] W. Yan, D. Tang, and Y. Lin, “A Data-Driven Soft Sensor Modeling Method Based on Deep Learning and its Application,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 5, pp. 4237–4245, May 2017, doi: 10.1109/TIE.2016.2622668.

[18] J. Moll, C. Kexel, J. Kathol, C. P. Fritzen, M. Moix-Bonet, C. Willberg, M. Rennoch, M. Koerd, and A. Herrmann, “Guided waves for damage detection in complex composite structures: The influence of omega stringer and different reference damage size,” *Applied Sciences (Switzerland)*, vol. 10, no. 9, May 2020, doi: 10.3390/app10093068.

[19] A. Bellini, A. Yazidi, F. Filippetti, C. Rossi, and G. A. Capolino, “High frequency resolution techniques for rotor fault detection of induction

machines," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 12, pp. 4200–4209, 2008, doi: 10.1109/TIE.2008.2007004.

[20] Agrawal Divyakant *et al.*, "Challenges and Opportunities with Big Data," 2011. [Online]. Available: http://docs.lib.purdue.edu/cctechhttp://docs.lib.purdue.edu/cctech/1

[21] E. Brusa, L. Cibrario, C. Delprete, and L. G. Di Maggio, "Explainable AI for Machine Fault Diagnosis: Understanding Features' Contribution in Machine Learning Models for Industrial Condition Monitoring," *Applied Sciences (Switzerland)*, vol. 13, Feb. 2023, doi: 10.3390/app13042038.

[22] M. H. Soleimani-Babakamali, R. Sepasdar, K. Nasrollahzadeh, I. Lourentzou, and R. Sarlo, "Toward a general unsupervised novelty detection framework in structural health monitoring," *Computer-Aided Civil and Infrastructure Engineering*, vol. 37, no. 9, pp. 1128–1145, Jul. 2022, doi: 10.1111/mice.12812.

[23] G. Sosale and J. Gebhardt, "Sensor Use Cases in the Context of Industry 4.0," in *Proceedings - 2020 2nd International Conference on Societal Automation, SA 2020*, Institute of Electrical and Electronics Engineers Inc., 2020. doi: 10.1109/SA51175.2021.9507167.

[24] IDG Research Services, "Studie Machine Learning / Deep Learning 2019." 2019.

[25] N. Helwig, E. Pignanelli, and A. Schütze, "Detecting and Compensating Sensor Faults in a Hydraulic Condition Monitoring System," in *Proceedings SENSOR 2015*, AMA Service GmbH, Jan. 2015, pp. 641–646. doi: 10.5162/sensor2015/D8.1.

[26] T. Schneider, N. Helwig, S. Klein, and A. Schütze, "Influence of Sensor Network Sampling Rate on Multivariate Statistical Condition Monitoring of Industrial Machines and Processes," in *EUROSENSORS 2018*, MDPI, Jan. 2018, p. 781. doi: 10.3390/proceedings2130781.

[27] ABB Motion Services, "ABB Ability Smart Sensors: Condition Monitoring für drehende Maschinen." 2020. [Online]. Available: https://library.e.abb.com/public/978686613d16469eb40b3f2ce04b7029/ABB%20Ability_Digital%20Powertrain_RevE_RZ_220309_web.pdf

[28] R. Vullers, R. Schaijk, H. Visser, J. Penders, and C. Hoof, "Energy Harvesting for Autonomous Wireless Sensor Networks," *IEEE Solid-State Circuits Magazine*, vol. 2, no. 2, pp. 29–38, 2010, doi: 10.1109/MSSC.2010.936667.

[29] G. Plastiras, M. Terzi, C. Kyrkou, and T. Theocharides, “Edge Intelligence: Challenges and Opportunities of Near-Sensor Machine Learning Applications,” in *2018 IEEE 29th International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, IEEE, Jul. 2018, pp. 1–7. doi: 10.1109/ASAP.2018.8445118.

[30] Bundesministerium für Bildung und Forschung, “KI-PREDICT.” Accessed: Feb. 07, 2024. [Online]. Available: https://www.elektronikforschung.de/projekte/ki-predict

[31] Bundesministerium für Bildung und Forschung, “KI-MUSIK4.0.” Accessed: Feb. 07, 2024. [Online]. Available: https://www.elektronikforschung.de/projekte/ki-musik4.0

[32] N. Helwig, “Zustandsbewertung industrieller Prozesse mittels multivariater Sensordatenanalyse am Beispiel hydraulischer und elektromechanischer Antriebssysteme,” Saarbrücken, 2018.

[33] Y.-H. H. Tsai and R. Salakhutdinov, “Improving One-Shot Learning through Fusing Side Information,” Oct. 2017, [Online]. Available: http://arxiv.org/abs/1710.08347

[34] R. Slatter and T. Schneider, “Influence of sensor network sampling rate on multivariate statistical condition monitoring of industrial machines and processes,” in *Praxisforum Elektrische Antriebstechnik 2019*, Jan. 2019.

[35] R. T. Olszewski, *Generalized Feature Extraction for Structural Pattern Recognition in Time-Series Data*. Carnegie Mellon University, 2001. doi: 10.5555/935627.

[36] N. Helwig, E. Pignanelli, and A. Schütze, “Condition monitoring of a complex hydraulic system using multivariate statistics,” in *2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, IEEE, Jan. 2015, pp. 210–215. doi: 10.1109/I2MTC.2015.7151267.

[37] S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis,” *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1–3, pp. 37–52, Jan. 1987, doi: 10.1016/0169-7439(87)80084-9.

[38] Case School of Engineering, “Case Western Reserve University Bearing Dataset,” *https://engineering.case.edu/bearingdatacenter*. [Online]. Available: https://engineering.case.edu/bearingdatacenter

[39] Case School of Engineering, “Case Western Reserve University Bearing Dataset Description,” *https://engineering.case.edu/bearingdatacenter/bearing-information*. [Online]. Available: https://engineering.case.edu/bearingdatacenter/bearing-information

[40] F. Mörchen, “Time series feature extraction for data mining using DWT and DFT,” in *Technical Report No. 33*, Dept. of Mathematics and Computer Science, University of Marburg, Germany, 2003.

[41] S. Klein, “Multivariate Auswertung spektralter Beschleunigungsdaten für das Condition Monitoring,” Bachelor-Thesis, Universität des Saarlandes, Saarbrücken, 2015.

[42] T. Schneider, “Methoden der automatisierten Merkmalextraktion und -selektion von Sensorsignalen,” Master-thesis, Saarland University, Saarbrücken, 2016.

[43] VDE Association for Electrical Electronic & Information Technologies, “Mikrosystemtechnik Kongress.” Ludwigsburg, 2021.

[44] N. V. Vapnik, *The Nature of Statistical Learning Theory*, Second Edition. New York: Springer science & business media, 1999.

[45] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, Jan. 2015, doi: 10.1038/nature14539.

[46] google.com, “Googel Scholar Citations of ‘Deep Learning’ by LeCun et al.” [Online]. Available: https://scholar.google.de/scholar?cites=5362332738201102290

[47] M. Hildebrand, M. Gebauer, and M. Mielke, *Daten- und Informationsqualität*, 5th ed. Springer Fachmedien Wiesbaden, 2021. doi: 10.1007/978-3-658-30991-6.

[48] F. Dekking, C. Kraaikamp, H. Lopuhaä, and L. Meester, *A Modern Introduction to Probability and Statistics: Understanding Why and How*. Springer, 2005.

[49] C. Meske, E. Bunde, J. Schneider, and M. Gersch, “Explainable Artificial Intelligence: Objectives, Stakeholders, and Future Research Opportunities,” *Information Systems Management*, vol. 39, no. 1, pp. 53–63, 2022, doi: 10.1080/10580530.2020.1849465.

[50] Sharpened Productions, “Windows Photos supported files.” Accessed: Apr. 18, 2022. [Online]. Available: https://fileinfo.com/software/microsoft/photos

[51] The MathWorks Inc., “Matlab Pretrained Neural Networks.” Accessed: Jan. 20, 2023. [Online]. Available:

https://de.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html

[52] B. S. Everitt and A. Skrondal, *The Cambridge Dictionary of Statistics*, 4th ed. Cambridge University Press, 2010.

[53] K. P. Burnham and D. Anderson, *Model Selection and Multimodel Inference*. Springer New York, 2004. doi: 10.1007/b97636.

[54] C. Böhm, S. Berchtold, and D. A. Keim, "Searching in high-dimensional spaces," *ACM Computing Surveys*, vol. 33, no. 3, pp. 322–373, Jan. 2001, doi: 10.1145/502807.502809.

[55] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When Is 'Nearest Neighbor' Meaningful?," *Database Theory—ICDT'99: 7th International Conference Jerusalem, Israel, January 10–12, 1999 Proceedings 7*, vol. 1540, Jan. 1997.

[56] P. Goodarzi, A. Schütze, and T. Schneider, "Comparing AutoML and Deep Learning Methods for Condition Monitoring using Realistic Validation Scenarios," Aug. 2023, [Online]. Available: http://arxiv.org/abs/2308.14632

[57] G. Serin, B. Sener, A. M. Ozbayoglu, and H. O. Unver, "Review of tool condition monitoring in machining and opportunities for deep learning," *The International Journal of Advanced Manufacturing Technology*, vol. 109, no. 3–4, pp. 953–974, Jan. 2020, doi: 10.1007/s00170-020-05449-w.

[58] A. Stetco, F. Dinmohammadi, X. Zhao, V. Robu, D. Flynn, M. Barnes, J. Keane, and G. Nenadic, "Machine learning methods for wind turbine condition monitoring: A review," *Renewable Energy*, vol. 133. Elsevier Ltd, pp. 620–635, Jan. 2019. doi: 10.1016/j.renene.2018.10.047.

[59] P. Li, F. Kong, Q. He, and Y. Liu, "Multiscale slope feature extraction for rotating machinery fault diagnosis using wavelet analysis," *Measurement: Journal of the International Measurement Confederation*, vol. 46, no. 1, pp. 497–505, 2013, doi: 10.1016/j.measurement.2012.08.007.

[60] H. H. Bafroui and A. Ohadi, "Application of wavelet energy and Shannon entropy for feature extraction in gearbox fault detection under varying speed conditions," *Neurocomputing*, vol. 133, pp. 437–445, Jan. 2014, doi: 10.1016/j.neucom.2013.12.018.

[61] D. Kateris, D. Moshou, X. E. Pantazi, I. Gravalos, N. Sawalhi, and S. Loutridis, "A machine learning approach for the condition monitoring of rotating

machinery," *Journal of Mechanical Science and Technology*, vol. 28, no. 1, pp. 61–71, Jan. 2014, doi: 10.1007/s12206-013-1102-y.

[62] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, "Time Series Analysis, Forecasting and Control." John Wiley & Sons, 2016. doi: 10.1057/9781137291264_6.

[63] B. Schlecht, "Wälzlager und Wälzlagerungen," *Maschienenelemente 2*. Pearson Studium, 2017.

[64] E. Keogh and S. Kasetty, "On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration," *Data Mining and Knowledge Discovery*, vol. 7, no. 4, pp. 349–371, Jan. 2003, doi: 10.1023/A:1024988512476.

[65] T. Wuest, D. Weimer, C. Irgens, and K.-D. Thoben, "Machine learning in manufacturing: advantages, challenges, and applications," *Production & Manufacturing Research*, vol. 4, no. 1, pp. 23–45, Jan. 2016, doi: 10.1080/21693277.2016.1192517.

[66] R. Agrawal, C. Faloutsos, and A. Swami, "Efficient similarity search in sequence databases," *Foundations of Data Organization and Algorithms*. pp. 69–84, 1993. doi: 10.1007/3-540-57301-1_5.

[67] H. A. Dau, E. Keogh, K. Kamgar, C.-C. M. Yeh, S. Z. Y. and Gharghabi, C. A. Ratanamahatana, Yanping, N. H. B. and Begum, A. Bagnall, A. Mueen, B. Gustavo, and Hexagon-ML, "The UCR Time Series Classification Archive." Jan. 2018. [Online]. Available: https://www.cs.ucr.edu/~eamonn/time_series_data_2018/

[68] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista, "The UCR Time Series Classification Archive." Jan. 2015. [Online]. Available: www.cs.ucr.edu/~eamonn/time_series_data/

[69] H. A. Dau, A. Bagnall, K. Kamgar, C. C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, and E. Keogh, "The UCR time series archive," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 6, pp. 1293–1305, Jan. 2019, doi: 10.1109/JAS.2019.1911747.

[70] X. Xi, E. Keogh, C. Shelton, L. Wei, and C. A. Ratanamahatana, "Fast time series classification using numerosity reduction," in *Proceedings of the 23rd international conference on Machine learning - ICML '06*, ACM Press, 2006, pp. 1033–1040. doi: 10.1145/1143844.1143974.

[71] A. Bostrom and A. Bagnall, “Binary Shapelet Transform for Multiclass Time Series Classification,” *Big Data Analytics and Knowledge Discovery*. Springer, pp. 257–269, 2015. doi: 10.1007/978-3-319-22729-0_20.

[72] J. Lines, S. Taylor, and A. Bagnall, “Time Series Classification with HIVE-COTE,” *ACM Transactions on Knowledge Discovery from Data*, vol. 12, no. 5, pp. 1–35, Jan. 2018, doi: 10.1145/3182382.

[73] J. Lines, S. Taylor, and A. Bagnall, “HIVE-COTE: The hierarchical vote collective of transformation-based ensembles for time series classification,” in *Proceedings - IEEE International Conference on Data Mining, ICDM*, Institute of Electrical and Electronics Engineers Inc., Jan. 2017, pp. 1041–1046. doi: 10.1109/ICDM.2016.74.

[74] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, “The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances,” *Data Mining and Knowledge Discovery*, vol. 31, no. 3, pp. 606–660, Jan. 2017, doi: 10.1007/s10618-016-0483-9.

[75] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. A. Muller, “Deep learning for time series classification: a review,” *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, Jan. 2019, doi: 10.1007/s10618-019-00619-1.

[76] J. Hills, J. Lines, E. Baranauskas, J. Mapp, and A. Bagnall, “Classification of time series by shapelet transformation,” *Data Mining and Knowledge Discovery*, vol. 28, no. 4, pp. 851–881, 2014, doi: 10.1007/s10618-013-0322-1.

[77] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, Jan. 2017, doi: 10.1145/3065386.

[78] I. Bilbao and J. Bilbao, “Overfitting problem and the over-training in the era of data: Particularly for Artificial Neural Networks,” in *2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS)*, IEEE, Jan. 2017, pp. 173–177. doi: 10.1109/INTELCIS.2017.8260032.

[79] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural Networks*, vol. 2, no. 5, pp. 359–366, Jan. 1989, doi: 10.1016/0893-6080(89)90020-8.

[80] K. Guo, W. Li, K. Zhong, Z. Zhu, S. Zeng, S. Han, Y. Xie, P. Debacker, M. Verhelst, and Y. Wang, “Neural Network Accelerator Comparison,” *https://nicsefc.ee.tsinghua.edu.cn/projects/neural-network-accelerator/*.

[Online]. Available: https://nicsefc.ee.tsinghua.edu.cn/projects/neural-network-accelerator/

[81] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," Sep. 2014, [Online]. Available: http://arxiv.org/abs/1409.4842

[82] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Computer Vision and Pattern Recognition*, Jan. 2015, doi: 10.48550/arXiv.1512.03385.

[83] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Jan. 2014, [Online]. Available: http://arxiv.org/abs/1409.1556

[84] Stanford Vision Lab, "ImageNet Database." [Online]. Available: https://www.image-net.org/index.php

[85] S. Pültz, "Automatisierte Methodenauswahl und Hyperparametertuning für Condition Monitoring mit maschinellem Lernen," Master-Thesis, Saarland University, 2021.

[86] R. Kohavi and D. H. Wolpert, "Bias Plus Variance Decomposition for Zero-One Loss Functions," in *International Conference on Machine Learning*, 1996, pp. 275–283.

[87] F. E. Harrell, K. L. Lee, R. M. Califf, D. B. Pryor, and R. A. Rosati, "Regression modelling strategies for improved prognostic prediction," *Statistics in Medicine*, vol. 3, no. 2, pp. 143–152, 1984, doi: 10.1002/sim.4780030207.

[88] G. v Trunk, "A Problem of Dimensionality: A Simple Example," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 3, pp. 306–307, 1979, doi: 10.1109/TPAMI.1979.4766926.

[89] R. A. Fisher, "The Statistical Utilization Of Multiple Measurements," *Annals of Eugenics*, vol. 8, no. 4, pp. 376–386, Aug. 1938, doi: 10.1111/j.1469-1809.1938.tb02189.x.

[90] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases," *Knowledge and Information Systems*, vol. 3, no. 3, pp. 263–286, Jan. 2001, doi: 10.1007/PL00011669.

[91] C. Faloutsos and V. Megalooikonomou, “On data mining, compression, and Kolmogorov complexity,” *Data Mining and Knowledge Discovery*, vol. 15, no. 1, pp. 3–20, Jan. 2007, doi: 10.1007/s10618-006-0057-3.

[92] E. J. Keogh and M. J. Pazzani, “An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback,” in *KDD’98: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, Jan. 1998, pp. 239–243. doi: 10.5555/3000292.3000335.

[93] I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh, *Feature Extraction: Foundations and Applications*, vol. 207. Springer, 2008.

[94] I. Guyon and A. Elisseeff, “An Introduction to Variable and Feature Selection,” *J. Machine Learning Research Special Issue on Variable and Feature Selection*, vol. 3, pp. 1157–1182, Jan. 2003, doi: 10.1162/153244303322753616.

[95] M. Espadoto, N. S. T. Hirata, and A. C. Telea, “Deep learning multidimensional projections,” *Information Visualization*, vol. 19, no. 3. SAGE Publications Ltd, pp. 247–269, Jan. 2020. doi: 10.1177/1473871620909485.

[96] T. G. Dietterich, “Machine Learning for Sequential Data: A Review,” *Structural, Syntactic, and Statistical Pattern Recognition*. Springer, pp. 15–30, 2002. doi: 10.1007/3-540-70659-3_2.

[97] D. Cai, X. He, and J. Han, “Training linear discriminant analysis in linear time,” in *Proceedings - International Conference on Data Engineering*, 2008, pp. 209–217. doi: 10.1109/ICDE.2008.4497429.

[98] G. J. McLachlan, “Mahalanobis distance,” *Resonance*, vol. 4, no. 6, pp. 20–26, Jan. 1999, doi: 10.1007/BF02834632.

[99] P. C. Mahalanobis, “On The Generalized Distance In Statistics,” in *Proceedings of the National Institute of Sciences of India*, 1936, pp. 49–55.

[100] P. Goodarzi, A. Schütze, and T. Schneider, “Comparison of different ML methods concerning prediction quality, domain adaptation and robustness,” *Technisches Messen*, vol. 89, no. 4, pp. 224–239, Apr. 2022, doi: 10.1515/teme-2021-0129.

[101] D. Luo, C. Ding, and H. Huang, “Linear Discriminant Analysis: New Formulations and Overfit Analysis,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 25, no. 1, pp. 417–422, Jan. 2011, doi: 10.1609/aaai.v25i1.7926.

[102] M. Leidinger, T. Sauerwald, W. Reimringer, G. Ventura, and A. Schütze, “Selective detection of hazardous VOCs for indoor air quality applications using a virtual gas sensor array,” *Journal of Sensors and Sensor Systems*, vol. 3, no. 2, pp. 253–263, Jan. 2014, doi: 10.5194/jsss-3-253-2014.

[103] P. Reimann and A. Schütze, “Fire detection in coal mines based on semiconductor gas sensors,” *Sensor Review*, vol. 32, no. 1. pp. 47–58, 2012. doi: 10.1108/02602281211197143.

[104] S. Wold, M. Sjöström, and L. Eriksson, “PLS-regression: a basic tool of chemometrics,” *Chemometrics and Intelligent Laboratory Systems*, vol. 58, no. 2, pp. 109–130, Jan. 2001, doi: 10.1016/S0169-7439(01)00155-1.

[105] M. Abadi *et al.*, “TensorFlow: A system for large-scale machine learning,” *Distributed, Parallel, and Cluster Computing*, p. 44, Jan. 2016, [Online]. Available: http://arxiv.org/abs/1605.08695

[106] google.com, “TensorFlow Lite.” [Online]. Available: https://www.tensorflow.org/lite

[107] A. Hennig, P. Gembaczka, L. Cousin, and A. Grabmaier, “Smart self-sufficient wireless current sensor,” in *Smart SysTech 2018; European Conference on Smart Objects, Systems and Technologies*, 2018, pp. 1–6.

[108] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, “XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9908, pp. VII–IX, Mar. 2016, [Online]. Available: http://arxiv.org/abs/1603.05279

[109] A. Liu, “The Application of Reconfigurable Computing in AI Chips,” in *Arm AI Gloabal Developers Conference*, 2019.

[110] R. Kozma, R. E. Pino, and G. E. Pazienza, “Are Memristors the Future of AI?,” *Advances in Neuromorphic Memristor Science and Applications*. Springer Netherlands, pp. 9–14, 2012. doi: 10.1007/978-94-007-4491-2_2.

[111] S. Markidis, S. W. Der Chien, E. Laure, I. B. Peng, and J. S. Vetter, “NVIDIA Tensor Core Programmability, Performance & Precision,” in *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, IEEE, May 2018, pp. 522–531. doi: 10.1109/IPDPSW.2018.00091.

[112] Intel Corporation, “Intel Movidius Vision Processing Units.” [Online]. Available:

https://www.intel.com/content/www/us/en/products/details/processors/movidius-vpu.html

[113] C. Marantos, N. Karavalakis, V. Leon, V. Tsoutsouras, K. Pekmestzi, and D. Soudris, “Efficient support vector machines implementation on Intel/Movidius Myriad 2,” in *2018 7th International Conference on Modern Circuits and Systems Technologies, MOCAST 2018*, Institute of Electrical and Electronics Engineers Inc., Jan. 2018, pp. 1–4. doi: 10.1109/MOCAST.2018.8376630.

[114] J. Schauer, A. Schütze, and T. Schneider, “Deep Neural Network Representation for Explainable Machine Learning Algorithms: A Method for Hardware Acceleration,” in *2024 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, IEEE [accepted, unpublished], 2024.

[115] L. Martí, N. Sanchez-Pi, J. M. Molina, and A. C. B. Garcia, “Anomaly detection based on sensor data in petroleum industry applications,” *Sensors (Switzerland)*, vol. 15, no. 2, pp. 2774–2797, Jan. 2015, doi: 10.3390/s150202774.

[116] J. Toivola, M. A. Prada, and J. Hollmén, “Novelty detection in projected spaces for structural health monitoring,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010, pp. 208–219. doi: 10.1007/978-3-642-13062-5_20.

[117] M. A. F. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, “A review of novelty detection,” *Signal Processing*, vol. 99. pp. 215–249, Jan. 2014. doi: 10.1016/j.sigpro.2013.12.026.

[118] T. Fawcett, “ROC Graphs: Notes and Practical Considerations for Data Mining Researchers,” *Machine Learning*, vol. 31, pp. 1–38, Jan. 2004.

[119] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, “Estimating the Support of a High-Dimensional Distribution,” *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, Jan. 2001, doi: 10.1162/089976601750264965.

[120] T. Dorst, “Measurement uncertainty in machine learning-uncertainty propagation and influence on performance,” doctoral dissertation, Saarland University, Saarbrücken, 2023.

[121] C. Schnur, S. Klein, and A. Blum, “Checklist Measurement and data planning for machine learning in assembly,” *ZeMA gGmbH*. Saarbrücken, 2023. doi: 10.5281/zenodo.7556875.

[122] P. Goodarzi, S. Klein, A. Schutze, and T. Schneider, “Comparing Different Feature Extraction Methods in Condition Monitoring Applications,” in *Conference Record - IEEE Instrumentation and Measurement Technology Conference*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/I2MTC53148.2023.10176106.

[123] Y. Robin, J. Amann, T. Schneider, A. Schütze, and C. Bur, “Comparison of Transfer Learning and Established Calibration Transfer Methods for Metal Oxide Semiconductor Gas Sensors,” *Atmosphere*, vol. 14, no. 7, Jul. 2023, doi: 10.3390/atmos14071123.

[124] Y. Robin, “The Potential of Deep Learning for Gas Sensor Evaluation and Calibration,” doctoral thesis [under review], Saarland University, Saarbrücken, Germany, 2024.

[125] S. Pültz, Y. Robin, Y. Koch, D. Q. Pais, L. Rauber, E. Kirchner, A. Schütze, and T. Schneider, “Automated Condition Monitoring for Helical Gears based on measuring Instantaneous Angular Speed with Magnetoresistive Sensors,” in *Sensors and Measuring Systems; 21th ITG/GMA-Symposium*, 2022, pp. 283–287.

[126] H. El Moutaouakil, J. Prager, A. Schütze, and T. Schneider, “Machine Learning Model Based on Signal Difference Features for Damage Localization on Hydrogen Pressure Vessel Using Ultrasonic Guided Waves,” in *Tagungsband 22. ITG/GMA-Fachtagung Sensoren und Messsysteme 2024*, AMA Science [accepted, unpublished], 2024.

[127] ZeMA gGmbH, “LMT ML Toolbox auf github.” [Online]. Available: https://github.com/ZeMA-gGmbH/LMT-ML-Toolbox

[128] ODION GmbH, “ODION Digital Factory.” [Online]. Available: https://odion.com/digital-factory/

[129] T. Schneider, “Project EaSy-ML at ZeMA.” Accessed: Nov. 29, 2023. [Online]. Available: https://zema.de/projekt/easy-ml/

[130] M. Bastuck, T. Baur, and A. Schütze, “DAV3E-a MATLAB toolbox for multivariate sensor data evaluation,” *Journal of Sensors and Sensor Systems*, vol. 7, no. 2, pp. 489–506, Jan. 2018, doi: 10.5194/jsss-7-489-2018.

[131] Saarland University Lab for Measurement Technology, “DAV^{3}E on github.” [Online]. Available: https://github.com/lmtUds/dav3e-beta

[132] C. Schnur, T. Dorst, K. Deshmukh, S. Zimmer, P. Litzenburger, T. Schneider, L. Margies, R. Müller, and A. Schütze, “PIA-A Concept for a Personal

Information Assistant for Data Analysis and Machine Learning of Time-Continuous Data in Industrial Applications," *ing.grid*, vol. 1, no. 2, pp. 1–7, 2023, doi: 10.48694/inggrid.3827.

[133] M. A. Gelbart, J. Snoek, and R. P. Adams, "Bayesian Optimization with Unknown Constraints," *arXiv preprint*, Jan. 2014, doi: 10.48550/arXiv.1403.5607.

[134] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-Adversarial Training of Neural Networks," *Journal of Machine Learning Research*, vol. 17, pp. 1–35, Jan. 2015, doi: 10.48550/arXiv.1505.07818.

[135] E. Hajiramezanali, S. Z. Dadaneh, A. Karbalayghareh, M. Zhou, and X. Qian, "Bayesian multi-domain learning for cancer subtype discovery from next-generation sequencing count data," *Advances in Neural Information Processing Systems*, vol. 31, Jan. 2018.

9 Used Tools

The following tools were used for spell- and grammar checking:

- Microsoft Word
- Grammarly Premium excluding AI prompt text generation (for examples see https://www.grammarly.com/features)

Appendix A: List of Publications

Peer-reviewed journal articles: 17 total, three as main author, four as scientific head:

- C. Schnur, T. Dorst, K.S. Deshmukh, S. Zimmer, P. Litzenburger, T. Schneider, L. Margies, R. Müller, and A. Schütze, „PIA - A Concept for a Personal Information Assistant for Data Analysis and Machine Learning of Time-Continuous Data in Industrial Applications“, in *ing.grid* 1(2), 2023 doi: 10.48694/inggrid.3827
- Y. Robin, J. Amann, T. Schneider, A. Schütze, and C. Bur, „Comparison of Transfer Learning and Established Calibration Transfer Methods for Metal Oxide Semiconductor Gas Sensors“ in *Atmosphere* 2023, 14(7), 1123, doi: 10.3390/atmos14071123
- T. Dorst, T. Schneider, S. Eichstädt, and A. Schütze, „Influence of measurement uncertainty on machine learning results demonstrated for a smart gas sensor“, in *J. Sens. Sens. Syst.*, 12, 45–60, 2023, doi: 10.5194/jsss-12-45-2023
- C. Schnur, S. Klein, A. Schütze, T. Schneider, and A. Blum, „Steigerung der Datenqualität in der Montage“, in *WT Werkstattstechnik*, 112 (2022) NR. 11-12, S. 783 - 787, doi: 10.37544/1436–4980–2022–11–12–57
- Y. Robin, J. Amann, P. Goodarzi, T. Schneider, A. Schütze, and C. Bur, „Deep Learning Based Calibration Time Reduction for MOS Gas Sensors with Transfer Learning“, in *Atmosphere* 2022, 13(10), 1614, doi: 10.3390/atmos13101614
- T. Dorst, T. Schneider, S. Eichstädt, and A. Schütze, „Uncertainty-aware automated machine learning toolbox (Automatisierte Toolbox für maschinelles Lernen unter Berücksichtigung von Messunsicherheiten)“, in *tm - Technisches Messen* (2023) 90(3), 141 - 153, doi: 10.1515/teme-2022-0042
- T. Dorst, M. Gruber, B. Seeger, A. P. Vedurmudi, T. Schneider, S. Eichstädt, and A. Schütze, „Uncertainty-aware data pipeline of calibrated MEMS sensors used for machine learning”, in *Measurement: Sensors* (2022), 22, 100376, doi: 10.1016/j.measen.2022.100376
- P. Goodarzi, A. Schütze, and T. Schneider, „Comparison of different ML methods concerning prediction quality, domain adaptation, and robustness” in *tm - Technisches Messen* (2022), vol. 89(4), 224-239, doi: 10.1515/teme-2021-0129

- C. Schnur, P. Goodarzi, Y. Lugovtsova, J. Bulling, J. Prager, K. Tschöke, J. Moll, A. Schütze and T. Schneider, „Towards Interpretable Machine Learning for Automated Damage Detection Based on Ultrasonic Guided Waves", in *Sensors* 2022, 22(1), 406, doi: 10.3390/s22010406
- Y. Robin, J. Amann, T. Baur, P. Goodarzi, C. Schultealbert, T. Schneider and A. Schütze, „High-Performance VOC Quantification for IAQ Monitoring Using Advanced Sensor Systems and Deep Learning", in *Atmosphere* (2021) 12(11), 1487, DOI 10.3390/atmos12111487
- A. Blum, S. Klein, K. Kühn, T. Schneider, A. Schütze, and R. Müller, „Inprozess-Dichtheitsprüfung in der Montage", in *wt Werkstattstechnik* online (2021), 9-2021, S. 650, DOI 10.37544/1436-4980-2021-09-75
- T. Dorst, Y. Robin, S. Eichstädt, A. Schütze, and T. Schneider, „Influence of synchronization within a sensor network on machine learning results", in *J. Sens. Sens. Syst.*, 10 (2021), pp. 233–245, DOI: 10.5194/jsss-10-233-2021
- T. Schneider, S. Klein, and A. Schütze, „Machine learning in industrial measurement technology for detection of known and unknown faults of equipment and sensors", *tm - Technisches Messen* (2019), 86 (11), 706–718, doi: 10.1515/teme-2019-0086
- T. Schneider, N. Helwig, and A. Schütze, „Industrial condition monitoring with smart sensors using automated feature extraction and selection", in *IOP Meas. Sci. Technol.* (2018) 29 094002, DOI: 10.1088/1361-6501/aad1d4
- A. Schütze, N. Helwig, and T. Schneider, „Sensors 4.0 - smart sensors and measurement technology enable Industry 4.0", in *J. Sens. Sens. Syst.* (2018) 7, 359-371, DOI: 10.5194/jsss-7-359-2018
- M. Schüler, T. Schneider, T. Sauerwald, and A. Schütze, „Impedance-based detection of HMDSO poisoning in metal oxide gas sensors", in *tm - Technisches Messen* (2017), 84(11), 697-705, doi.org/10.1515/teme-2017-0002
- T. Schneider, N. Helwig, and A. Schütze, „Automatic feature extraction and selection for classification of cyclical time series data", in *tm - Technisches Messen* (2017), 84(3), 198-206, doi: 10.1515/teme-2016-0072

Conference contributions: 38 total, nine as main author, seven as scientific head

Authors	Title	Source	Year
E. Holle, F. Knödl, M. Mayer, T. Schneider, D. Spiehl, A. Blaeser, E. Dörsam, A. Schütze	Control of ink-water balance in offset lithography by machine learning	49th Conference of iarigai, oral presentation, September 18-20,	2023

Authors	Title	Source	Year
		2023, Wuppertal, Germany	
T. Schneider, T. Dorst, C. Schnur, P. Goodarzi, A. Schütze	Einfluss von Datenqualität, Domain Shift und Messunsicherheit auf die Vorhersagequalität smarter Sensorsysteme (Influence of data quality, domain shift, and measurement uncertainty on the prediction quality of smart sensor systems)	tm – Technisches Messen, 2023, 90(S1), 33-36, doi: 10.1515/teme-2023-0087	2023
C. Schnur, Y. Robin, P. Goodarzi, T. Dorst, A. Schütze, T. Schneider	Development of a bearing test-bed for acquiring data for robust and transferable machine learning	IEEE I2MTC 2023, International Instrumentation and Measurement Technology Conference, May 22 - 25, 2023, Kuala Lumpur, Malaysia	2023
P. Goodarzi, S. Klein, A. Schütze, T. Schneider	Comparing Different Feature Extraction Methods in Condition Monitoring Applications	IEEE I2MTC 2023, International Instrumentation and Measurement Technology Conference, May 22 - 25, 2023, Kuala Lumpur, Malaysia	2023
Y. Robin, J. Amann, P. Goodarzi, T. Schneider, A. Schütze, C. Bur	Comparison of Explainable Machine Learning Algorithms for Optimization of Virtual Gas Sensor Arrays	IEEE I2MTC 2023, International Instrumentation and Measurement Technology Conference, May	2023

Authors	Title	Source	Year
		22 - 25, 2023, Kuala Lumpur, Malaysia	
Tizian Schneider	Condition monitoring and process control with magnetic sensors and machine learning	16. XMR-Symposium "Magnetoresistive Sensors and Magnetic Systems", Wetzlar, March 8-9, 2023	2023
Christian Fuchs, Steffen Klein, Payman Goodarzi, Andreas Schütze, Tizian Schneider	Analyse zum Einfluss von Labeling-Fehlern im Kontext von Luftschall- und Vibrationsdatensätzen für maschinelles Lernen	DAGA 2023 - 49. Jahrestagung für Akustik, Tagungsband, S. 80-83, Sitzung "Akustische Messtechnik und Sensorik 1", Hamburg, 06.-09. März 2023	2023
Christopher Schnur, Steffen Klein, Anne Blum, Tizian Schneider, Rainer Müller und Andreas Schütze	Mess- und Datenplanung für Modelle des maschinellen Lernens an Bestandsanlagen	16. Dresdner Sensor-Symposium, Posterbeitrag, Dresden, 5.-7.12.2022	2022
Yannick Robin, Jannis Morsch, Tizian Schneider, Andreas Schütze, Christian Bur	Insight in Dynamically Operated Gas Sensor Arrays with Shapley Values for Data Segments	MNE EUROSENSORS 2022, Poster T3-P2-WeA_0, Leuven, BE, Sep. 19-23. 2022	2022
S. Pültz, Y. Robin, A. Schütze, T. Schneider,	Automated Condition Monitoring for Helical Gears	Vortrag, Sensoren und Messsysteme	2022

Authors	Title	Source	Year
Y. Koch, E. Kirchner, D. Quirnheim Pais, L. Rauber	based on measuring Instantaneous Angular Speed with Magnetoresistive Sensors	2022, Tagungsband, S. 283-287	
P. Goodarzi, A. Schütze, T. Schneider	Prediction quality, domain adaptation and robustness of Machine Learning methods: a comparison	Vortrag, Sensoren und Messsysteme 2022, Tagungsband, S. 281-282	2022
Christian Fuchs, Steffen Klein, Stefan Saller, Daniel Spies, Andreas Schütze, Tizian Schneider	Entwicklung akustischer Messungen für industrielles maschinelles Lernen	DAGA 2022 - 48. Jahrestagung für Akustik, Vorkolloquium „Künstliche Intelligenz für akustische Sensorsysteme“, 21. - 24. März 2022	2022
Yannick Robin, Johannes Amann, Payman Goodarzi, Tobias Baur, Caroline Schultealbert, Tizian Schneider, Andreas Schütze	Überwachung der Luftqualität in Innenräumen mittels komplexer Sensorsysteme und Deep Learning Ansätzen	15. Dresdner Sensor-Symposium, 6. - 8. Dezember 2021, Online Event, Vortrag, Session Smarte (Gas-)Sensorik	2021
C. Schnur, J. Moll, Y. Lugovstova, A. Schütze, T. Schneider	Explainable Machine Learning for Damage Detection in Carbon Fiber Composite Plates Under Varying Temperature Conditions	QNDE 2021 – 48th Annual Review of Progress in Quantitative Nondestructive Evaluation, July 28-30, 2021	2021

Authors	Title	Source	Year
T. Baur, C. Schultealbert, Y. Robin, P. Goodarzi, T. Schneider, and A. Schütze	Accurate Quantification of Formaldehyde at ppb Level for Indoor Air Quality Monitoring	IMCS 2021, International Meeting on Chemical Sensors, Digital Conference, Presentation IMCS 05-1576, May 30 - June 3, 2021	2021
Tanja Dorst, Yannick Robin, Tizian Schneider and Andreas Schütze	Automated ML Toolbox for Cyclic Sensor Data	Joint Virtual Workshop of ENBIS and MATHMET Mathematical and Statistical Methods for Metrology MSMM 2021, 31 May – 1 June 2021	2021
Yannick Robin, Payman Goodarzi, Tobias Baur, Caroline Schultealbert, Andreas Schütze, Tizian Schneider	Machine Learning-Based Calibration Time Reduction for Gas Sensors in Temperature Cycled Operation	IEEE I2MTC 2021, International Instrumentation and Measurement Technology Conference, Digital Conference, Session ´Sensors, Instrumentation & AI for Environmental Measurement´, May 17-20, 2021	2021
Tanja Dorst, Sascha Eichstädt, Tizian Schneider, Andreas Schütze	GUM2ALA – Uncertainty propagation algorithm for the Adaptive Linear Approximation according to the GUM	Oral presentation D1.1, SMSI 2021, Sensor and Measurement Science	2021

Authors	Title	Source	Year
		International, Digital Conference, 3 - 6 May 2021, DOI: 10.5162/SMSI2021/D1.1	
T. Dorst, S. Eichstädt, T. Schneider, A. Schütze	Propagation of uncertainty for an Adaptive Linear Approximation algorithm	SMSI 2020 - Measurement Science, pp 366 - 367, doi: 10.5162/SMSI2020/E2.3	2020
T. Dorst, S. Eichstädt, T. Schneider, A. Schütze	Metrology for the Factory of the Future: Entwicklung und Erweiterung metrologischer Standards für die digitale Fabrik der Zukunft	Last Minute Poster, 14. Dresdner Sensor-Symposium, Dresden, 2.-4. Dezember 2019	2019
Tizian Schneider	Machine Learning in der industriellen Messtechnik zur Erkennung bekannter und unbekannter Anlagen- und Sensorfehler	Vortrag, VDI/VDE-GMA-Expertenforum Trends in der Mess- und Automatisierungstechnik – Von der Messung zur Information, Karlsruhe, 28./29. November 2019	2019
T. Dorst, T. Schneider, S. Klein, S. Eichstädt, A. Schütze	Influence of synchronization within a sensor system on machine learning results	oral presentation, MATHMET 2019 International Workshop, Lisbon	2019

Authors	Title	Source	Year
		(Portugal), 20–22 November 2019	
A. Schütze, S. Klein, T. Dorst, T. Schneider	Sensorik 4.0 – smarte Sensorsysteme ermöglichen Zustandsbewertung, Selbstüberwachung und Prozessoptimierung	Vortrag, Jahrestreffen der ProcessNet-Fachgemeinschaften "Prozess-, Apparate- und Anlagentechnik" unterstützt durch „Sustainable Production, Energy and Resources", 4.-5. November 2019, Dortmund	2019
Tizian Schneider, Steffen Klein, Anne Blum, Leonie Schirmer, Rainer Müller, Andreas Schütze	Combination of Human and Machine Intelligence to Optimize Assembly	Societal Automation - Technological & Architectural Frameworks, Krakow (Poland), 4-6 September 2019	2019
T. Dorst, T. Schneider, S. Klein, S. Eichstädt, A. Schütze	Synchronisationsprobleme innerhalb eines Sensorsystems und deren Auswirkungen auf Ergebnisse des maschinellen Lernens	20. GMA/ITG Fachtagung Sensoren und Messsysteme 2019, Nürnberg, 25. und 26. Juni 2019	2019
S. Klein, T. Schneider, A. Schütze	Zustandsüberwachung in der Automatisierungstechnik mittels maschinellem Lernen	20. GMA/ITG Fachtagung Sensoren und Messsysteme 2019,	2019

Authors	Title	Source	Year
		Nürnberg, 25. und 26. Juni 2019	
Tanja Dorst, Björn Ludwig, Sascha Eichstädt, Tizian Schneider, Andreas Schütze	Metrology for the factory of the future: towards a case study in condition monitoring	IEEE I2MTC 2019 International Instrumentation and Measurement Technology Conference, May 20 - 23, 2019, Auckland, New Zealand	2019
Tizian Schneider, Nikolai Helwig, Steffen Klein, Andreas Schütze	Influence of sensor network sampling rate on multivariate statistical condition monitoring of industrial machines and processes	EUROSENSORS 2018, poster presentation, September 9-12, 2018, Graz, Austria.	2018
T. Schneider, S. Klein, N. Helwig, A. Schütze, M. Selke, C. Nienhaus, D. Laumann, M. Siegwart, K. Kühn	Big Data Analytik mit automatisierter Signalverarbeitung für Condition Monitoring	Sensoren und Messsysteme 2018, 19. ITG/GMA-Fachtagung, Vortrag, Session Sensorik für die Industrie 4.0, 26. - 27. Juni 2018, Nürnberg, D. in: ITG-Fachbericht 281: Sensoren und Messsysteme, VDE-Verlag Berlin (2018), ISBN 978-3-8007-4683-5, S. 259-262.	2018
T. Schneider, N. Helwig, A. Schütze	Automatic Feature Extraction and Selection for Condition	I^2MTC-2018 - The IEEE 2018	2018

Authors	Title	Source	Year
	Monitoring and related Datasets	International Instrumentation and Measurement Technology Conference, poster, Proc. pp. 429-434 (ISBN: 978-1-5386-2222-3), May 14-17, Houston, TX, USA.	
Tizian Schneider, Nikolai Helwig, Andreas Schütze	Modular Sensor Systems for real-time Process Control and Smart Condition Monitoring - MoSeS-Pro	IEEE Sensors Conference 2017, open poster, Glasgow, Scotland, Oct 30 - Nov 1, 2017.	2017
Nikolai Helwig, Philip Merten, Tizian Schneider, Andreas Schütze	Integrated Sensor System for Condition Monitoring of Electromechanical Cylinders	MDPI Proceedings 2017, 1, 626, Proc. EUROSENSORS 2017, Paris, France, Sep. 3-6, 2017.	2017
N. Helwig, T. Schneider, A. Schütze	Modular sensor systems for real-time process control and smart condition monitoring using XMR technology	14th xMR-Symposium "Magnetoresistive Sensors and Magnetic Systems", Sensitec GmbH (ed.), Wetzlar, Germany, March 21-22, 2017. proceedings page 15-22."	2017

Authors	Title	Source	Year
M. Bastuck, T. Baur, T. Schneider, A. Schütze	DAV³E - a comprehensive toolbox for multisensor data fusion not only for gas sensors	Proceedings, 6th EuNetAir scientific meeting, Prague, Oct. 05-07, 2016.	2016
N. Helwig, T. Schneider, A. Schütze	Modulare Sensorsysteme für Echtzeit-Prozesssteuerung und smarte Zustandsbewertung	Vortrag, VDI-Fachkonferenz Intelligente Sensoren für Industrie 4.0, Nürtingen, Germany, 20.-21. September 2016	2016
Tizian Schneider, Nikolai Helwig, Andreas Schütze	Automatic feature extraction and selection for classification of cyclical time series data	Vortrag, XXX. Messtechnisches Symposium des AHMT, Hannover, 15.-16. September 2016. in: Stefan Zimmermann (Hrsg.): Tagungsband des XXX. Messtechnischen Symposium 2016, De Gruyter, ISBN 978-3-11-049487-7	2016

Invited Talks:

Authors	Title	Source	Year
Tizian Schneider	KI für KMU	AMA Mitgliederversammlung 2023, eingeladener Vortrag,	2023

Authors	Title	Source	Year
		Nürnberg, 10.05.2023	
Tizian Schneider	Impulsvortrag Trainingsdaten für industrielles Machine-Learning	Digitale Fachkonferenz "Forschung für Edge Computing 2023", 24. Januar 2023	2023
Tizian Schneider	Zustandsüberwachung der Mechatronik durch schnelle Stromsensoren und Algorithmen zur automatisierten Merkmalextraktion und Fehlerklassifikation	Innovationsplattform Magnetische Mikrosysteme INNOMAG e.V., Mitgliedertreffen, Mainz, 05. Dezember 2019	2019

Datasets:

Authors	Title	Source	Year
Tizian Schneider, Steffen Klein, Manuel Bastuck	Condition monitoring of hydraulic systems Data Set at ZeMA	dataset published on Zenodo, April 26, 2018	2018
Tizian Schneider, Steffen Klein, Manuel Bastuck	Condition monitoring of hydraulic systems Data Set	dataset published in UCI machine learning repository, April 26, 2018	2018

Appendix B: List of Projects in DESS Group

Table 1 lists the projects conducted in the DESS group and the author's contribution.

Table 1: Research, transfer, and teaching projects that emerged in the context of the toolbox developed by the author totaling 4,079,614€ in third-party research funding.

Project name (classification, start-end, budget DESS)	Content for group DESS	Author's contribution
MessMo – Measurement aided assembly (research; 04/2018-09/2020; 352,183 €)	Extension of the toolbox's scope to assembly processes (previously wear and fabrication)	Contribution to the project proposal and consulting on the project
EaSy-ML - Evaluation Assistance System for Machine Learning (transfer; 03/2019-02/2021; 92,228 €)	Toolbox integration with data acquisition software for Odion GmbH	Started as a scientific staff, later as a project manager
Met4FoF - Metrology for Factories of the Future (research; 06/2018-09/2021; 132,500 €)	Metrological framework for estimation of measurement uncertainty of the toolbox according to GUM	As head of group DESS
KomZet Saar - mittelstand4.0 competence center Saarbrücken (transfer; 09/2017-08/2022; 47,596 €)	Consulting, sensibilization, training, and networking for small and medium businesses on digitalization and AI	Participation as AI trainer, coordination of DESS contributions, contribution to the project proposal extension
KI-Predict - Electronics for on-edge condition monitoring with distributed AI (research; 03/2020-12/2023; 718,200 €)	Specialization of the toolbox to magnetic field and vibration sensors as well as support for an implementation on an FPGA and an ASIC (inference only)	Significant contribution to the project proposal and project manager for UdS
KI-MUSIK4.0 – microelectronics-based universal sensor interface for AI in Industry 4.0 (research; 04/2020-03/2023; 537,628 €)	Specialization of the toolbox to microphones, a significant simplification of the employed algorithms for application on low-cost inference ASICs	Significant contribution to the project proposal and project manager for UdS
Magie-KI – AI Monitoring and control of color-water-balance for offset printing (research; 10/2021-09/2023; 230,388 €)	Data analysis on data from gas and humidity sensors for closed-loop control of color-water-balance in offset printing	As head of group DESS

BetoNPP – hybrid measurement approach to monitoring and surveying thick-layered reinforced concrete structures in nuclear plants (research; 05/2021-12/2024; 273,298 €)	Sensor fusion of georadar and leakage flux for localization and assessment of reinforcing steel in the concrete of up to one-meter thickness, as well as automated documentation	Responsible for project proposal, project manager for UdS
ITec-Pro – Research and development of innovative processes and technologies for the production of the future (research; 06/2021-08/2022; 108,379 €)	Guideline for data and measurement planning for machine learning projects in small and medium businesses, as well as a personal assistant program for analysis	Contribution to the project proposal and as head of group DESS
Pre-Project Edge-Power - Robust and secure edge electronics for industrial processes and critical infrastructure (research; 10/2020-06/2021; 69,987 €)	Estimation of potential energy savings of intelligent, energy-self-sufficient edge computing modules for condition monitoring vs. cloud computing	As head of group DESS
Edge-Power - Robust and secure edge electronics for industrial processes and critical infrastructure (research; 07/2022-08/2025; 452,793 €)	Research on concepts for distributed training of ML algorithms on energy-self-sufficient edge hardware as well as their dimensioning	Significant contribution to the project proposal and as head of group DESS
DDMI – Digital data management for engineering (teaching; 06/22-11/2023; 19,500 €)	Teaching of data and project management by employing modern teaching and learning methods	Conception, coordination, and preparation of "Digital Data Management for Engineering Sciences"
MDZ - Mittelstand digitalization center (transfer; 09/2022-08/2025; 110,584 €)	Workshops and consulting for small and medium businesses on ML and AI	Proposal (part of the DESS group), coordination of the contributions of the DESS group
KI-Mono - AI for monitoring hydrogen pressure vessels with ultrasonic-guided waves (research; 10/2022-09/2025; 258,237 €)	Structural health monitoring of hydrogen pressure vessels by combining the toolbox and ultrasonic guided waves.	Project proposal and project manager UdS
VProSaar – Distributed production for the automotive industry in Saarland: sustainable, connected, resilient (research; 10/2022-09/2026; 458,028 €)	Research on commonly encountered domain shifts, transferability issues of ML, and concepts for increased robustness against domain shifts	Project proposal, coordination of contributions of the DESS group

NFDI4Ing – National Research Data Infrastructure for Engineering Sciences (research 10/2020-09/2025, 50.500€)	Research data infrastructure and systematic implementation of FAIR data principles	As head of group DESS