Measurement uncertainty in machine learning – uncertainty propagation and influence on performance

Dissertation
zur Erlangung des Grades
der Doktorin der Ingenieurswissenschaften
der Naturwissenschaftlich-Technischen Fakultät
der Universität des Saarlandes

von
Tanja Dorst

Saarbrücken
2023

Tag des Kolloquiums:	14. Juli 2023
Dekan:	Prof. Dr. Ludger Santen
Berichterstatter:	Prof. Dr. Andreas Schütze
	Prof. Dr.-Ing. Rainer Tutsch
Vorsitz:	Prof. Dr. Romanus Dyczij-Edlinger
Akad. Mitarbeiter:	Dr.-Ing. Amine Othmane

"But in my opinion, all things in nature occur mathematically."

René Descartes

Abstract

Industry 4.0 is based on the intelligent networking of machines and processes in industry and makes a decisive contribution to increasing competitiveness. For this, reliable measurements of used sensors and sensor systems are essential. Metrology deals with the definition of internationally accepted measurement units and standards. In order to internationally compare measurement results, the *Guide to the Expression of Uncertainty in Measurement* (GUM) provides the basis for evaluating and interpreting measurement uncertainty. At the same time, measurement uncertainty also provides data quality information, which is important when machine learning is applied in the digitalized factory. However, measurement uncertainty in line with the GUM has been mostly neglected in machine learning or only estimated by cross-validation.

Therefore, this dissertation aims to combine measurement uncertainty based on the principles of the GUM and machine learning. For performing machine learning, a data pipeline that fuses raw data from different measurement systems and determines measurement uncertainties from dynamic calibration information is presented. Furthermore, a previously published automated toolbox for machine learning is extended to include uncertainty propagation based on the GUM and its supplements. Using this uncertainty-aware toolbox, the influence of measurement uncertainty on machine learning results is investigated, and approaches to improve these results are discussed.

Zusammenfassung

Industrie 4.0 basiert auf der intelligenten Vernetzung von Maschinen und Prozessen und trägt zur Steigerung der Wettbewerbsfähigkeit entscheidend bei. Zuverlässige Messungen der eingesetzten Sensoren und Sensorsysteme sind dabei unerlässlich. Die Metrologie befasst sich mit der Festlegung international anerkannter Maßeinheiten und Standards. Um Messergebnisse international zu vergleichen, stellt der *Guide to the Expression of Uncertainty in Measurement* (GUM) die Basis zur Bewertung von Messunsicherheit bereit. Gleichzeitig liefert die Messunsicherheit auch Informationen zur Datenqualität, welche wiederum wichtig ist, wenn maschinelles Lernen in der digitalisierten Fabrik zur Anwendung kommt. Bisher wurde die Messunsicherheit im Bereich des maschinellen Lernens jedoch meist vernachlässigt oder nur mittels Kreuzvalidierung geschätzt.

Ziel dieser Dissertation ist es daher, Messunsicherheit basierend auf dem GUM und maschinelles Lernen zu vereinen. Zur Durchführung des maschinellen Lernens wird eine Datenpipeline vorgestellt, welche Rohdaten verschiedener Messsysteme fusioniert und Messunsicherheiten aus dynamischen Kalibrierinformationen bestimmt. Des Weiteren wird eine bereits publizierte automatisierte Toolbox für maschinelles Lernen um Unsicherheitsfortpflanzungen nach dem GUM erweitert. Unter Verwendung dieser Toolbox werden der Einfluss der Messunsicherheit auf die Ergebnisse des maschinellen Lernens untersucht und Ansätze zur Verbesserung dieser Ergebnisse aufgezeigt.

Appended Papers

Paper 1 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, *Measurement: Sensors* (2022)

Paper 2 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 3 T. Dorst, T. Schneider, S. Eichstädt, and A. Schütze: Uncertainty-aware automated machine learning toolbox, *tm - Technisches Messen* (2023)

Paper 4 T. Dorst, T. Schneider, S. Eichstädt, and A. Schütze: Influence of measurement uncertainty on machine learning results demonstrated for a smart gas sensor, *Journal of Sensors and Sensor Systems* (2023)

Author's contribution to appended papers

Paper 1 I developed the concept and the methodology, implemented the software, carried out the analysis, investigated the results, and wrote the original draft of the paper.

Paper 2 I carried out the time shift analysis, visualized the results, and wrote the original draft of the paper.

Paper 3 I carried out the derivation of the formulas, analysis, visualized the results, and wrote the original draft of the paper.

Paper 4 I carried out the analysis, visualized the results, and wrote the original draft of the paper.

Other Related Publications

Data set A T. Dorst, M. Gruber, and A. P. Vedurmudi: Sensor data set of one electromechanical cylinder at ZeMA testbed (ZeMA DAQ and Smart-Up Unit), *Zenodo* (2021)

Paper A T. Dorst, B. Ludwig, S. Eichstädt, T. Schneider, and A. Schütze: Metrology for the factory of the future: towards a case study in condition monitoring, *IEEE I2MTC 2019 International Instrumentation and Measurement Technology Conference* (2019)

Paper B T. Dorst, T. Schneider, S. Klein, S. Eichstädt, and A. Schütze: Synchronisationsprobleme innerhalb eines Sensorsystems und deren Auswirkungen auf Ergebnisse des maschinellen Lernens, *20. GMA/ITG Fachtagung Sensoren und Messsysteme* (2019)

Paper C T. Dorst, S. Eichstädt, T. Schneider, and A. Schütze: Propagation of uncertainty for an Adaptive Linear Approximation algorithm, *SMSI 2020 – Sensor and Measurement Science International* (2020)

Paper D T. Dorst, Y. Robin, T. Schneider, and A. Schütze: Automated ML Toolbox for Cyclic Sensor Data, *Joint Virtual Workshop of ENBIS and MATHMET Mathematical and Statistical Methods for Metrology MSMM 2021* (2021)

Paper E T. Dorst, P. Christoffel, J. Kunze, and A. Schütze: Abschätzung der Messunsicherheit in smarten Sensornetzwerken mittels künstlichem Rauschen und maschinellem Lernen, *15. Dresdner Sensor-Symposium* (2021)

Paper F M. Gruber, T. Dorst, A. Schütze, S. Eichstädt, and C. Elster: Discrete wavelet transform on uncertain data: Efficient online implementation for practical applications, in *Advanced Mathematical and Computational Tools in Metrology and Testing XII* (2022)

Paper G T. Dorst, M. Gruber, A. P. Vedurmudi, D. Hutzschenreuter, S. Eichstädt, and A. Schütze: Providing FAIR and metrologically traceable data sets - a case study, *IMEKO TC6 International Conference on Metrology and Digital Transformation* (2022)

Paper H T. Dorst, M. Gruber, A. P. Vedurmudi, D. Hutzschenreuter, S. Eichstädt, and A. Schütze: A Case Study on providing FAIR and metrologically traceable data sets, *Acta IMEKO* (2023)

Table of Contents

List of Figures

List of Tables

List of Listings

List of Abbreviations

$D4$	Daubechies-4
ADC	analog-to-digital converter
AI	artificial intelligence
AIoT	Artificial Intelligence of Things
ALA	Adaptive Linear Approximation
AMLT	automated machine learning toolbox
ANN	artificial neural network
BDW	Best Daubechies Wavelets
BFC	Best Fourier Coefficients
CC-BY-4.0	Creative Commons Attribution 4.0 International
CO_2	carbon dioxide
CV	cross-validation
CWT	Continuous Wavelet Transform
DAQ	data acquisition unit
dB	decibel
DC	Dublin Core
DFT	Discrete Fourier Transform
DL	deep learning
DOI	Digital Object Identifier
DSI	Digital System of Units
DUT	device under test
DWT	Discrete Wavelet Transform
EMC	electromechanical cylinder
EMPIR	European Metrology Programme for Innovation and Research
FAIR	Findable, Accessible, Interoperable, Reusable
FE	feature extraction
FoF	Factory of the Future
FS	feature selection
GMA	gas mixing apparatus
GNSS	Global Navigation Satellite System

GPS	Global Positioning System
GUM	Guide to the Expression of Uncertainty in Measurement
GUM-S1	Supplement 1 to the GUM
GUM-S2	Supplement 2 to the GUM
HDF5	Hierarchical Data Format Version 5
I4.0	Industry 4.0
IAQ	indoor air quality
IIoT	Industrial Internet of Things
IIR	infinite impulse response
IoT	Internet of Things
JCGM	Joint Committee for Guides in Metrology
JSON	JavaScript Object Notation
KNN	K-nearest neighbors
LASSO	Least Absolute Shrinkage and Selection Operator
LDA	Linear Discriminant Analysis
LHS	Latin Hypercube Sampling
LPU	Law of Propagation of Uncertainty
MAE	mean absolute error
MCM	Monte Carlo method
MEMS	micro-electro-mechanical systems
Met4FoF	Metrology for the Factory of the Future
ML	machine learning
MOS	metal oxide semiconductor
NFL	No Free Lunch Theorem
PC	principal component
PCA	Principal Component Analysis
PDF	probability density function
PLS	Partial Least Square
PLSR	Partial Least Squares Regression
ppb	parts per billion
ppm	parts per million
PPS	pulse-per-second
PTB	Physikalisch-Technische Bundesanstalt

QUDT	Quantities, Units, Dimensions, and Types
RDF	Resource Description Framework
RFELSR	Recursive Feature Elimination Least Squares Regression
RFESVM	Recursive Feature Elimination Support Vector Machine
RFESVR	Recursive Feature Elimination Support Vector Regression
RH	relative humidity
RMS	root-mean-square
RMSE	root-mean-square error
RReliefF	Regressional ReliefF
RUL	remaining useful lifetime
SI	Système international d'unités
SIMPLS	statistically inspired modification of the PLS
SM	Statistical Moments
SNR	signal-to-noise ratio
SOSA	Sensor, Observation, Sample, and Actuator
SSN	Semantic Sensor Network
SUU	Smart-Up Unit
SVD	Singular Value Decomposition
SVM	Support Vector Machine
$\mathrm{T+U-RMSE}$	test plus uncertainty RMSE
$\mathrm{T-RMSE}$	test RMSE
TC	temperature cycle
TCO	temperature cycled operation
TCOCNN	10-layer deep convolutional neural network
TLV	threshold limit value
$\mathrm{U-RMSE}$	uncertainty RMSE
UA-AMLT	uncertainty-aware automated machine learning toolbox
UGM	unique gas mixture
VOC	volatile organic compound
$\mathrm{VOC_{sum}}$	sum of VOC concentrations
W3C	World Wide Web Consortium
ZeMA	Zentrum für Mechatronik und Automatisierungstechnik gGmbH

List of Symbols

$\mathbf{D}$	cyclic data matrix
$\mathbf{F_E}$	matrix containing extracted features
$\mathbf{F_S}$	matrix containing the optimum number of selected features
$g_{X_i}(\xi_i)$	probability density function for X_i
I	identity matrix
j	imaginary unit $\sqrt{-1}$
$\mathrm{lb}(a)$	binary logarithm of a, logarithm to the base 2 of a
μ	mean value
$\mathbb{N}$	natural numbers with zero
$\mathbb{N}_{>0}$	natural numbers without zero
$\mathcal{N}(\mu, \sigma^2)$	Gaussian distribution with parameters μ and σ^2
$\mathbb{R}^n$	n-dimensional space of real numbers
$\mathbb{R}_{\geq 0}$	non-negative real numbers
$r_{Pearson}$	Pearson correlation coefficient
σ	standard deviation
s_{ALA}	number of determined segments in the ALA algorithm
s_{BDW}	number of chosen wavelet coefficients in the BDW algorithm
s_{BFC}	number of chosen amplitudes / phases in the BFC algorithm
s_{PCA}	number of chosen principal components in the PCA algorithm
s_{SM}	number of segments in the SM algorithm
$u(x_i)$	standard uncertainty associate with x_i
$u_c(y)$	combined standard uncertainty associate with y
$\mathbf{U_{F_E}}$	uncertainty matrix associated with matrix $\mathbf{F_E}$
$\mathbf{U_{F_S}}$	uncertainty matrix associated with matrix $\mathbf{F_S}$
$\mathbf{U_x}$	covariance matrix associated with $\mathbf{x}$
X_i	input quantity

x_i	estimate of the input X_i
$\mathbf{X}$	matrix
$\mathbf{X}^{-1}$	invertible matrix of $\mathbf{X}$
$\mathbf{X}^\top$	transpose of matrix $\mathbf{X}$
$\mathbf{x}$	vector
$\mathbf{x}^\top$	transposed vector
Y	measurand
y	estimate of the measurand Y
$\mathbf{y}$	target vector
$\widehat{\mathbf{y}}$	predicted target vector

1 Introduction

This chapter provides a short introduction to metrology and artificial intelligence (AI). Furthermore, it describes the scope of the dissertation and gives a brief outline of its chapter structure.

1.1 Motivation

Metrology is the "science of measurement and its application" [1]. It deals with the definition of internationally accepted measurement units, the realization of these measurement units in practice, as well as the establishment of traceability by linking practical measurements with reference standards using calibration [2].

Metrological traceability is crucial to ensure that measurements carried out at different times, at different locations, by different measurement systems, and by different engineers are comparable. The need for measurement standards and traceability dates back to the French Revolution. At this time, a uniform system of units was required for trading between different countries. In 1793, René Just Haüy developed the first document that established relations between different national measures [3]. The *Système international d'unités* (SI) [4] used today is based on the metric system that was adopted by the French government during the French Revolution [5]. The latest change was made on May 20, 2019, when the kilogram is no longer defined by a cylinder of platinum-iridium alloy (Le Grand K) but, from then on, is defined by the Planck constant [6]. In addition to an internationally accepted metric system, the precision levels of the various national industries and the associated measurement uncertainty also play an essential role in global trade. As all measurements are subject to uncertainty, a measurement result is denoted as complete only if it contains both the measured value and the associated measurement uncertainty as a quantitative statement on the quality. To compare measurement results worldwide, the *Guide to the Expression of Uncertainty in Measurement* (GUM) [7] was developed by the members of the *Joint Committee for Guides in Metrology* (JCGM) in 1993. It provides internationally accepted rules

for evaluating and expressing measurement uncertainty. Nowadays, the GUM and its supplements define the de facto standard for uncertainty evaluation in the field of metrology.

In the age of the fourth industrial revolution, also called *Industry 4.0* (I4.0), the digitalization of factories is one of the most important decisive factors for increasing competitiveness and efficiency [8]. Optimizing production processes and driving technological advancements will enhance quality, productivity, and flexibility while reducing human errors and costs in a digitalized factory, the so-called *Factory of the Future* (FoF). The *Industrial Internet of Things* (IIoT), as a subset of the *Internet of Things* (IoT), uses interconnected smart sensors, actuators, and devices to enhance production processes, increase efficiency, and improve safety and health [9]. New applications are emerging for analyzing large amounts of collected data, allowing automation and improvement of complex processes, leading to the field of AI with its important subset, machine learning (ML) (cf. Figure 1.1). Combining AI technologies with IoT infrastructure leads to *Artificial Intelligence of Things* (AIoT), a current trend in smart industries [10]. In addition to the application in industry, e.g., to perform predictive maintenance, AI is used in a wide field of applications, e.g., in internet search engines (e.g., Google) or in self-driving cars [11].

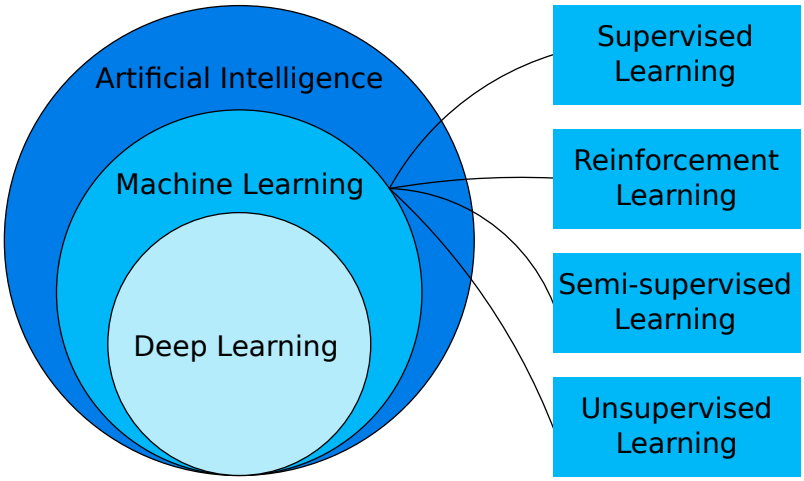


Figure 1.1: Relationship between artificial intelligence and its subfields, machine learning (with its different learning types), as well as deep learning (adapted from [12]).

The founding of AI as a field of research dates back to the Dartmouth Conference in 1956 [13]. AI broadly describes the approach of using machines to solve problems by imitating intelligent human behavior, i.e., thinking and acting like humans [14–16].

There are many definitions for AI, but no single correct working one exists, as shown in [17]. ML, a subset of AI, is the general term for computers learning from data alone without being explicitly programmed [18, 19]. In the field of deep learning (DL), a form of ML involving the use of multilayer artificial neural networks (ANNs), algorithms adjust themselves to increase accuracy without any human intervention [20]. An ANN is an information processing technique with a layered structure inspired by the biological structure of the human brain [21]. ANNs essentially are ML algorithms, and therefore, DL is a subset of ML; however, comparing DL with ML brings up some differences. While ML uses chiefly structured data, i.e., data that can be expressed in rows and columns, DL can also deal with unstructured data, e.g., audio, image, and text [20, 22]. Unstructured data means the desired information is unavailable in a structured form, e.g., if a dog is visible in an image [23]. DL achieves better accuracy when large amounts of data are available, whereas ML is better at using smaller data sets [24]. A benefit of using ML instead of DL is the better interpretability of the models and their decisions [25].

This dissertation aims to bring both fields together, metrology on the one hand and ML on the other, leading to uncertainty-aware ML. The focus of this dissertation is set on ML model building and the reliability of the models' decisions and predictions by considering measurement uncertainty in the entire ML process. When decisions or predictions are based on ML models, confidence in the used algorithms is essential, and therefore, reliable measurement data and the assessment of data quality are required as reliability is directly affected by data quality. Data quality can be expressed in terms of measurement uncertainty, whose evaluation is essential for metrological traceability.

However, evaluating the uncertainty of each specific prediction of an ML model is often neglected so far [26], and the average performance of the ML model is only assessed by cross-validation (CV) [27]. The typically used k-fold CV splits the data set into $k \in \mathbb{N}$ subsets, trains the model with $k-1$ subsets, and uses the remaining subset for the model application to asses how accurately the model performs on data that has not been used for the training [28, 29]. Performing CV leads to uncertainty evaluation of the average ML model performance by using quantities such as the mean absolute error (MAE) or the root-mean-square error (RMSE) [26]. Quantifying the uncertainty of each specific prediction needs additional computational effort to that used for model training. Nevertheless, uncertainty quantification should not be considered as a burden but as a worthwhile addition leading to reliable ML predictions.

The *European Metrology Programme for Innovation and Research* (EMPIR) project *Metrology for the Factory of the Future* (Met4FoF) aims to provide a framework for the entire lifecycle of measured data in industrial applications, including traceable calibration of smart sensors, metrology in complex sensor networks, and uncertainty quantification in ML [30, 31]. By combining metrology for digital sensors, sensor networks, and data analysis performed by ML, the traceability chain becomes digitally enabled [31]. In sensor networks, the key elements in the FoF, automated data transfer, and data analysis plays an important role. To achieve interoperability and automate data analysis in an I4.0 environment, raw sensor data must be enriched with machine-readable and machine-interpretable information in the form of metadata based on ontologies [32, 33] such as *Quantities, Units, Dimensions, and Types* (QUDT) [34] and *Semantic Sensor Network* (SSN) [35].

To achieve a digitally enabled traceability chain, smart sensors must provide uncertainty associated with the raw data [36] that can be used for ML. To evaluate the uncertainty of each specific prediction, uncertainty propagation through feature extraction (FE) and feature selection (FS) algorithms, which prepare the data for using ML algorithms, as well as the propagation through the used data-driven ML algorithm is required. The uncertainty propagation in the ML process is one specific challenge addressed in this dissertation. Many algorithms for FE, FS, classification, and regression exist, and they could all be extended with uncertainty propagation in line with the GUM and its supplements. However, this would be time-consuming, and thus, the decision in this dissertation was made to use an already published software toolbox for ML [37, 38], the so-called automated machine learning toolbox (AMLT), explained in detail in Section 2.2, which consists of a choice of complementary algorithm covering a wider field of applications. It has already been applied successfully to various data sets covering use cases such as industrial condition monitoring [39]. The AMLT consists of five FE and three FS algorithms as well as one classifier leading to a manageable effort to develop uncertainty propagation for each algorithm to make the AMLT uncertainty-aware. For some ML methods, uncertainty propagation was already developed, for example, for *discrete Fourier transform* (DFT) [40] and *discrete Wavelet transform* (DWT) [41, 42], so these algorithms require only minor adjustments. By extending all algorithms used in the AMLT with uncertainty propagation, the extended version of the AMLT can be used to perform uncertainty-aware ML. As this uncertainty-aware automated machine learning toolbox (UA-AMLT) is only suitable for classification problems, there is a

need for an adapted UA-AMLT for regression problems, including the corresponding uncertainty propagation, which is also addressed in this dissertation.

Using the developed UA-AMLT, the influence of measurement uncertainty on the ML performance can be investigated as a further challenge in this dissertation. As measurement uncertainty occurs in time (e.g., caused by timing issues in a sensor network) and value (e.g., caused by noise), both occurrences and their influences on the ML models and their performance are investigated.

1.2 Organization

After motivating this dissertation, including the reasons for this research and the objectives to be achieved, Chapter 2 introduces the fundamentals for the demonstrated developments and investigations. First, the measurement uncertainty according to the GUM and the used ML algorithms are presented, which will be extended by the measurement uncertainty propagation in the course of the thesis. For investigating the influence of measurement uncertainty on ML results, different data sets are presented:

- a data set of a lifetime test of an electromechanical cylinder,
- a data set of different calibration and field test measurements of gas mixtures with a metal oxide semiconductor (MOS) gas sensor, and
- a data set of a hydraulic system with different simulated failures.

This dissertation summarizes the work published in Papers 1 to 4. In Chapter 3, the papers themselves are introduced, starting in Section 3.1 with an overview of the relation between the paper presented in the following sections. Section 3.2 (Paper 1) deals with an uncertainty-aware data pipeline for calibrated micro-electro-mechanical systems (MEMS) sensors used for machine learning, including data alignment of two different data acquisition units (DAQs) and obtaining uncertainty values for raw data from calibration information. As an extension to Paper 1, Section 3.3 presents how data used in this paper can be made FAIR, i.e., findable, accessible, interoperable, and reusable, and the FAIRness level of this new data set is assessed. Subsequently, Section 3.4 (Paper 2) investigates the influence of time synchronization problems within a sensor network on ML results, and ways to improve the obtained results are discussed. In Section 3.5 (Paper 3), the development of an uncertainty-aware AMLT is presented by extending an existing AMLT with uncertainty propagation for every step (FE, FS, as well as

classification) within this toolbox. Section 3.6 (Paper 4) extends the uncertainty-aware AMLT for regression problems by developing an uncertainty-aware *Partial Least Squares Regression* (PLSR) version. Furthermore, the influence of measurement uncertainty on regression problem results is investigated, and potentials to optimize the overall measurement system, including ML, is shown.

This thesis is closed with a summary and an outlook on further research directions.

2 Fundamentals

This chapter covers the fundamentals of this thesis. It briefly introduces measurement uncertainty propagation in line with the *Guide to the Expression of Uncertainty in Measurement* (GUM) [7] and its supplements, Supplement 1 (GUM-S1) [43] and Supplement 2 (GUM-S2) [44]. In addition, some common machine learning methods, which will be referred to in the next chapter, are presented. Three data sets are used in this thesis to investigate the influence of measurement uncertainty on machine learning (ML) results. These data sets are explained in this chapter. To perform machine learning well and reproducible, data sets that fit FAIR principles are needed at best [45]. This chapter explains these principles and how a data set can achieve FAIRness, as well as a method to assess the FAIRness level of a data set.

2.1 Measurement Uncertainty

Measurement uncertainty is defined as a "non-negative parameter characterizing the dispersion of the quantity values being attributed to a measurand, based on the information used" [1, 2.26]. In practice, measurement uncertainties cannot be prevented, although their magnitude can be minimized by good measurement systems as well as appropriate processing and analysis of data. Common sources of measurement uncertainty are, for example, ambient conditions, characteristics of the measurement device as well as the human itself [7]. Figure 2.1 gives an overview of possible sources that potentially cause measurement uncertainty. Thus, uncertainty quantification from measurement setup, measurements themselves, to data evaluation in machine learning is essential. A framework for evaluating and expressing measurement uncertainty is given by the GUM [7], and its supplements GUM-S1 [43] and GUM-S2 [44]. With this framework, measurements and their derived quantities are evaluated according to a uniform process, and the resulting measurement accuracy, as well as the measurement results, are transparent, interpretable, and comparable worldwide.

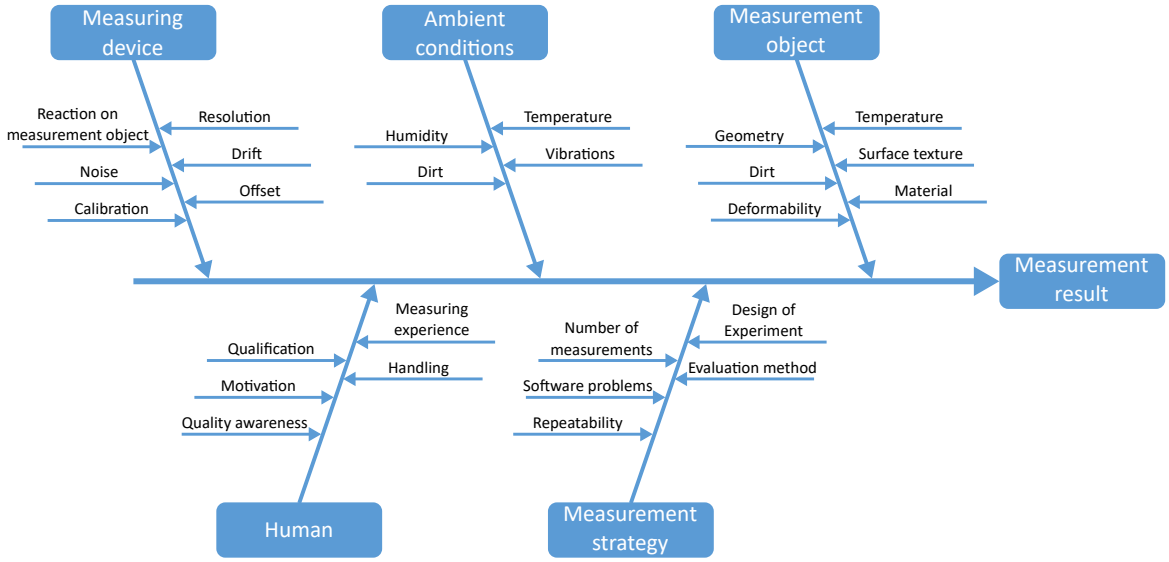


Figure 2.1: Ishikawa diagram showing potential causes of measurement uncertainty.

In accordance with the GUM, the uniform process of measurement uncertainty estimation consists of four main steps [7]:

1. Specification of the measurand Y.

2. Identification of the input quantities $X_1, X_2, \ldots, X_n$, which influence the measurement and determination of the standard uncertainty $u(x_i)$ for each of the input estimates x_i.

3. Development of a mathematical model

$$Y = f(\mathbf{X}) \tag{2.1}$$

with real-value input quantities $\mathbf{X} = (X_1, X_2, \ldots, X_n)^\top$, which expresses the functional dependence of the measurand Y on the input quantities X_i.

4. Calculation of the combined standard uncertainty $u_c(y)$, which is assigned to the estimated value of the measurand y.

In the GUM, a first-order Taylor series approximation, i.e., a linearization of the model equation $Y = f(\mathbf{X})$, is used to combine the individual standard uncertainties $u(x_i)$. The combined standard uncertainty of the measurement result is the positive square root of the estimated variance $u_c^2(y)$ obtained from the *Law of Propagation of*

Uncertainty (LPU) [7]

$$u_c^2(y) = \sum_{i=1}^{n} \left(\frac{\partial f}{\partial x_i} \right)^2 u^2(x_i) + \underbrace{2 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{\partial f}{\partial x_i} \frac{\partial f}{\partial x_j} u(x_i, x_j)}_{=0, \text{ if uncorrelated input quantities}} . \tag{2.2}$$

The sensitivity coefficients $c_i = \frac{\partial f}{\partial x_i}$ in Equation (2.2) describe how a small change in the input estimates $x_1, x_2, \ldots, x_n$ influences the estimated value of the measurand y.

With the sensitivity vector containing the derivatives with respect to the input estimates

$$\mathbf{f_x} = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n} \right) \tag{2.3}$$

and with the covariance matrix $\mathbf{U_x}$ of dimension $n \times n$ associated with $\mathbf{x}$

$$\mathbf{U_x} = \begin{pmatrix} u^2(x_1) & u(x_1, x_2) & \ldots & u(x_1, x_n) \\ u(x_2, x_1) & u^2(x_2) & \ldots & u(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ u(x_n, x_1) & u(x_n, x_2) & \ldots & u^2(x_n) \end{pmatrix}, \tag{2.4}$$

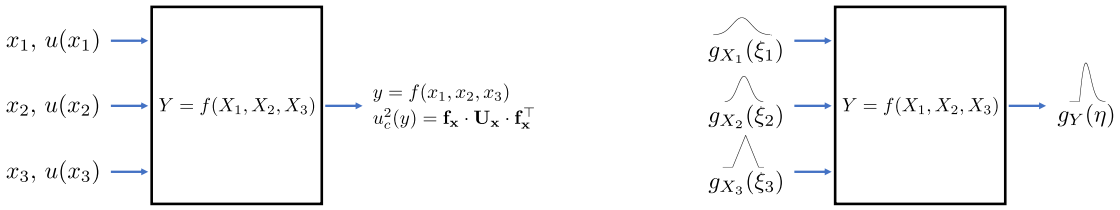
Equation (2.2) can be written in matrix-vector notation as

$$u_c^2(y) = \mathbf{f_x} \cdot \mathbf{U_x} \cdot \mathbf{f_x}^\top . \tag{2.5}$$

An illustration of the LPU is shown in Figure 2.2a.

In the classical GUM method described above, problems arise if, for example, the influence of the input quantities $X_1, X_2, \ldots, X_n$ on the measurand Y is described by a significantly nonlinear function or the input quantities are not normally but arbitrarily distributed (Central Limit Theorem, [7, G.2]). A more general approach without these limitations is presented in GUM-S1 [43], which deals with the propagation of probability distributions based on a Monte Carlo method (MCM). The probability density functions (PDFs) $g_{X_i}(\xi_i)$, $i = 1, \ldots, n$, for the input quantities X_i are propagated through the model in Equation (2.1) to obtain the PDF $g_Y(\eta)$ for the measurand Y. As this method is entirely numerical, it is difficult to identify the most significant contribution to the combined standard uncertainty. The computational cost strongly depends on the number of Monte Carlo trials. Figure 2.2b illustrates the propagation of probability distributions through a mathematical model using three independent input quantities

as an example. In this figure, $g_{X_1}(\xi_1)$ and $g_{X_2}(\xi_2)$ are PDFs of a Gaussian distribution $\mathcal{N}(\mu, \sigma^2 = u^2(x))$ with different values for σ, and $g_{X_3}(\xi_3)$ is a PDF of a triangular distribution. Propagating these three PDFs through the model leads to an asymmetric PDF $g_Y(\eta)$ for the measurand Y.



(a) Law of Propagation of Uncertainty (LPU).

(b) Propagation of probability density functions (PDFs).

Figure 2.2: Illustration of GUM and GUM-S1 for $n = 3$ input quantities (adapted from [43]).

The linearization method (GUM) and the Monte Carlo method (GUM-S1) can only be used if the input quantities are real-valued and if there is only a single scalar output quantity in the measurement model. Thus, Supplement 2 of the GUM extends both methods to complex-valued quantities and any number of output quantities. In this case, the measurement model can be written as

$$\mathbf{Y} = f(\mathbf{X}), \tag{2.6}$$

where $\mathbf{X} = (X_1, X_2, \ldots, X_n)^\top$ and $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_m)^\top$ denote the input and the output quantities, respectively.

Let $\mathbf{C_x}$ denote the sensitivity matrix of dimension $m \times n$ given by

$$\mathbf{C_x} = \begin{pmatrix} \dfrac{\partial f_1}{\partial x_1} & \cdots & \dfrac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial f_m}{\partial x_1} & \cdots & \dfrac{\partial f_m}{\partial x_n} \end{pmatrix}. \tag{2.7}$$

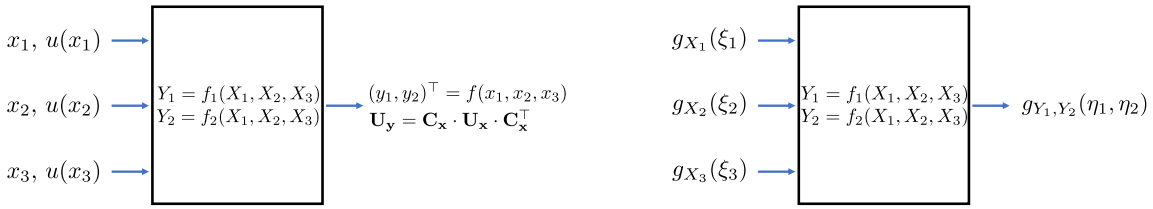
The covariance matrix of dimension $m \times m$ associated with $\mathbf{y}$ is given by

$$\mathbf{U_y} = \begin{pmatrix} u^2(y_1) & u(y_1, y_2) & \ldots & u(y_1, y_m) \\ u(y_2, y_1) & u^2(y_2) & \ldots & u(y_2, y_m) \\ \vdots & \vdots & \ddots & \vdots \\ u(y_m, y_1) & u(y_m, y_2) & \ldots & u^2(y_m) \end{pmatrix} \tag{2.8}$$

and calculated with

$$\mathbf{U_y} = \mathbf{C_x} \cdot \mathbf{U_x} \cdot \mathbf{C_x}^\top \tag{2.9}$$

according to [44, 6.2.1]. Figure 2.3 illustrates the LPU as well as the propagation of probability distributions through a mathematical model using three independent input and two output quantities as an example.



(a) Law of Propagation of Uncertainty (LPU).

(b) Propagation of probability density functions (PDFs).

Figure 2.3: Illustration of GUM-S2 for $n = 3$ mutually independent input quantities and $m = 2$ output quantities (adapted from [44]).

2.2 Machine Learning

When physics-based model building is impossible due to a lack of theoretical knowledge about the physical system or analytical model building is too complex, data-driven models based on ML can be used instead. The four main types of ML techniques are shown in Figure 2.4.

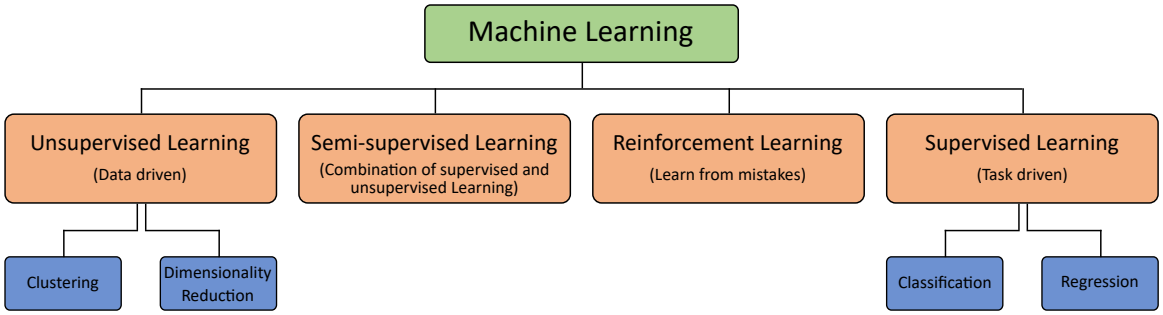


Figure 2.4: Main types of machine learning techniques and their algorithm types (adapted from [46]).

Unsupervised learning means finding hidden structures or patterns in unlabeled data [47]. Semi-supervised learning is a combination of unsupervised and supervised

learning, which uses labeled as well as unlabeled data [47]. Reinforcement learning is neither supervised nor unsupervised, as it is only based on rewarding desired and punishing undesired behaviors [23]. Supervised learning techniques, i.e., classification and regression, which are focused on in this thesis, use data $\mathbf{D}$ as input and a corresponding target vector $\mathbf{y}$ as output to learn the mapping function f from the input to the output. Then, it holds $\mathbf{y} = f(\mathbf{D})$. The difference between classification and regression is that the target $\mathbf{y}$ for classification is discrete (categorical) while it is continuous (numerical) for regression.

In [37], an automated machine learning toolbox (AMLT) for classification using the commercial software MATLAB® is presented, which neither needs any analytical model of the task at hand nor requires expert knowledge about data science. It can cope with a large variety of supervised learning problems using observations in the form of cyclic sensor data, i.e., at least one parameter in the measurements repeats a recurring pattern over time. If no cyclic sensor data is available, windowing can be used to data streams in the preprocessing step to split the unbounded data stream into finite ranges. Thus, cyclic data for one sensor is given by a matrix $\mathbf{D} \in \mathbb{R}^{m \times n}$, where m denotes the number of cycles and n is the number of samples per cycle. Using $c \in \mathbb{N}$ different sensors means that there are c different cyclic data matrices $\mathbf{D}_c \in \mathbb{R}^{m \times n_c}$, one for each sensor. The number of samples n_c for each matrix depends on the sampling rates of the sensors and can be different for all sensors but has to be the same for all cycles of one sensor. In the following, all mathematical considerations are shown on one cyclic data matrix $\mathbf{D}$ for one sensor for which it holds

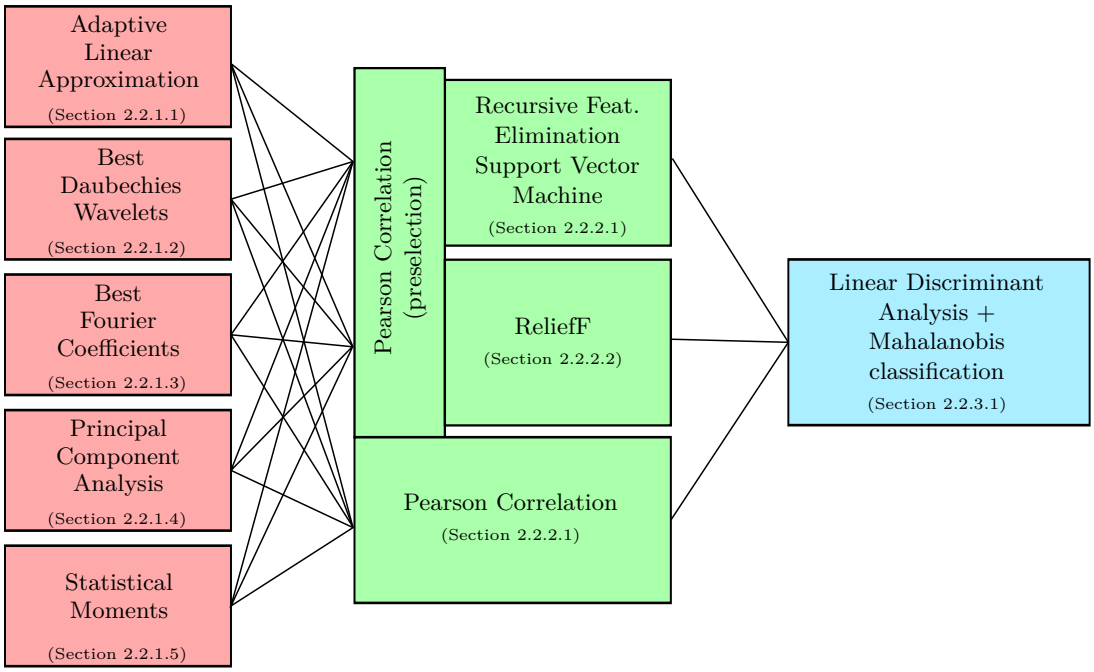
$$\mathbf{D} = \begin{pmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{m1} & d_{m2} & \dots & d_{mn} \end{pmatrix}. \tag{2.10}$$

The data matrix is often unsuitable for performing classification or regression directly due to its high dimensionality and redundancy. High-dimensional data $\mathbf{D}$ can lead to some counterintuitive mathematical effects, such as the exponential increase in volume when adding additional dimensions to a mathematical space [48], which are subsumed as the *curse of dimensionality* in the context of ML [49–52]. To avoid these mathematical effects, the dimensionality of the data must be reduced as much as possible without losing important information within the data. Moreover, multicollinearity, i.e., redundancy in the data, is a common problem in data analysis as the covariance matrix of the data

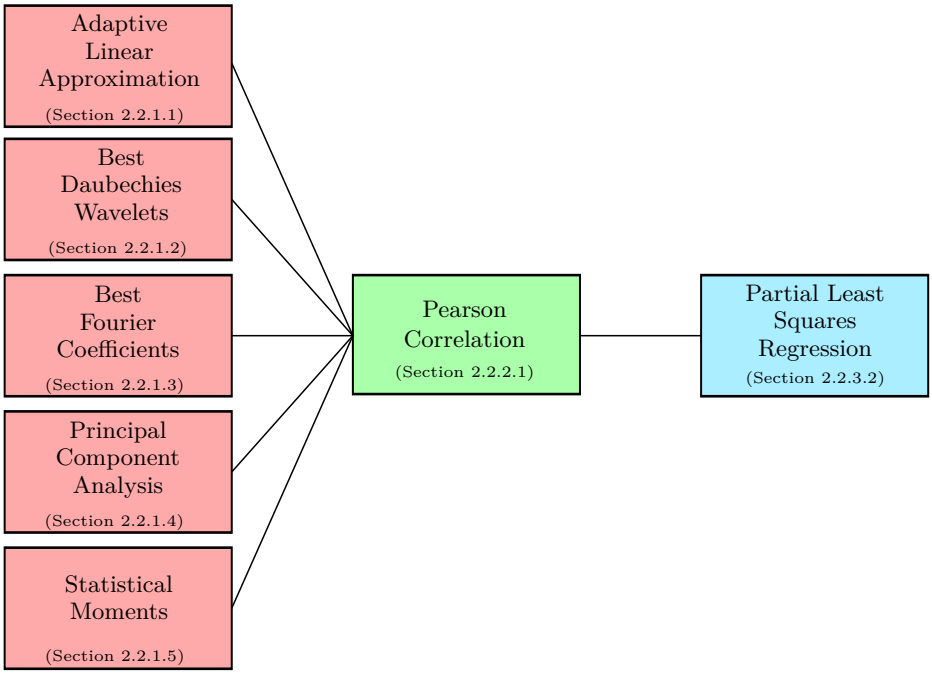
becomes singular and, therefore, the matrix is noninvertible. The singularity leads to numerical issues in various ML techniques, e.g., *Linear Discriminant Analysis* (LDA). For these two reasons, dimensionality reduction is necessary, which can be achieved by feature extraction (FE) (cf. Section 2.2.1) and if the number of features is still too high by subsequent feature selection (FS) (cf. Section 2.2.2). In addition, analyzing dimensionality-reduced data is less computationally expensive and better manageable [53]. Thus, the AMLT for classification problems consists of three main parts: FE, FS, and classification.

According to the *No Free Lunch Theorem* (NFL) [54–56], there is no single best algorithm for all classification problems. Therefore, the AMLT consists of several complementary algorithms for FE and FS. In [57], 14 FE and 66 FS algorithms have been investigated for classification problems. 49 of the 66 FS algorithms have been implemented and tested on several data sets, which leads to a combination of three complementary FS algorithms (cf. Section 2.2.2), resulting in good feature sets for every tested data set. Using this combination, 14 FE algorithms have been tested on nine data sets, including, among others, gas sensor and condition monitoring data. Five complementary FE algorithms, which reliably provide good results, have been chosen for the AMLT. These algorithms are explained in more detail in Section 2.2.1. Combining the five FE and the three FS algorithms leads to 15 combinations for the model training, where the best one, i.e., the combination with minimum cross-validation (CV) error (explained in Section 2.2.4), is used for the model application. The AMLT has been tested on several data sets from different use cases for classification tasks, e.g., industrial condition monitoring, activity recognition using smartphones, and day recognition using traffic information [39]. The AMLT did not fail for any tested data set, and the results were similar to or better than previously achieved results with other approaches, which are often specially designed for a specific data set [39]. A scheme of the AMLT is shown in Figure 2.5a. Using the AMLT, a mathematically optimal solution will not be reached in most cases, but a good trade-off solution for a broad range of use cases can be covered.

For regression problems, the AMLT for classification is slightly modified, as shown in Figure 2.5b. No FE algorithm is changed, but only one of the three FS algorithms can be used in the AMLT for regression problems as FS is supervised and only one algorithm can deal with continuous target values. Five FE algorithms and one FS algorithm result in only five possible algorithm combinations. The best combination is determined by the minimum CV error (with root-mean-square error (RMSE) as error measure) and later used for the model application.



(a) Scheme of the automated machine learning toolbox for classification (adapted from [38]).



(b) Scheme of the automated machine learning toolbox for regression (adapted from Paper 4, [58]).

Figure 2.5: Scheme of the automated machine learning toolbox for (a) classification (blue) and (b) regression (blue) together with feature extraction (red) and feature selection (green).

To perform ML with the AMLT, data must be preprocessed to obtain an equidistant and cyclic structure. This is one of the preprocessing steps included in the uncertainty-aware data pipeline for calibrated MEMS sensors shown in Paper 1 (Section 3.2). To perform uncertainty-aware ML, measurement uncertainty must be considered. Thus, Paper 1 (Section 3.2) presents how measurement uncertainty for raw data can be determined from calibration information. As the AMLT assumes equidistant timestamps due to the matrix representation, measurement uncertainty in time and the influence of ML results are considered in Paper 2 (Section 3.4). The extension of the AMLT to include consideration of measurement uncertainty in the values is based on the GUM [7] and its supplements, GUM-S1 [43] and GUM-S2 [44]. For classification problems, the extension is presented in Paper 3 (Section 3.5), and for regression problems, it is shown in Paper 4 (Section 3.6).

2.2.1 Feature Extraction

FE is an unsupervised step in the AMLT with the objective of concentrating as much information of the cyclic data set $\mathbf{D}$ in as few features as possible. In case of small data sets, FE is not necessarily a dimensionality reduction step as, in this case, the number of extracted features per cycle, denoted as k, can be larger than n, i.e., the number of samples per cycle. Five complementary algorithms are used in the AMLT to extract features from time, frequency, and time-frequency domain resulting in five feature sets containing mostly a high number of features. These five algorithms reliably provide good results on all tested data sets by Schneider [57]. For each of these algorithms, FE can be mathematically defined for data of one sensor as a mapping $\mathbf{D} \mapsto \mathbf{F_E}$, where $\mathbf{F_E} \in \mathbb{R}^{m \times k}$, $k \in \mathbb{N}$, denotes the matrix containing extracted features.

2.2.1.1 Adaptive Linear Approximation

Information contained in local details, like edges in the time domain, can be extracted using *Adaptive Linear Approximation* (ALA) [37]. ALA splits a measurement cycle into linear segments of variable length. As features, the mean value and the slope of every linear segment are extracted from the time domain [59]. Splitting a cycle into a large number of segments leads to many features and, thus, a small approximation error. In contrast, fewer features, i.e., fewer segments, leads to a larger approximation error. The number of segments s_{ALA} is calculated automatically by using data of all cycles

contained in $\mathbf{D}$ and stopping the splitting of the cycles when the approximation error does not significantly decrease with a further split. Thus, all cycles are split equally.

Let $\mathbf{d}_i = (d_{i1}, d_{i2}, \ldots, d_{in}) \in \mathbb{R}^{1\times n}$ denote the real-valued time-domain signal, i.e., the i-th measurement cycle of $\mathbf{D}$. As the calculations for every segment are the same, they are shown here only for the k-th segment of $\mathbf{d}_i$ with start index v_k and the end index v_{k+1}. A measurement value d_{ij} at time t_j, which lies within the k-th segment, can be linearly approximated by

$$d_{ij} = f_{ik}(t_j) = a_{ik} + b_{ik} \cdot (t_j - \overline{t_k}), \tag{2.11}$$

where a_{ik} denotes the mean value and b_{ik} the slope of the k-th segment within the i-th cycle. This segment's mean and slope can be calculated according to

$$a_{ik} = \frac{1}{v_{k+1} - v_k + 1} \sum_{j=v_k}^{v_{k+1}} d_{ij} \qquad \text{and} \tag{2.12}$$

$$b_{ik} = \frac{\sum_{j=v_k}^{v_{k+1}} (t_j - \overline{t_k})(d_{ij} - a_{ik})}{\sum_{j=v_k}^{v_{k+1}} (t_j - \overline{t_k})^2} \qquad \text{with} \tag{2.13}$$

$$\overline{t_k} = \frac{1}{v_{k+1} - v_k + 1} \sum_{j=v_k}^{v_{k+1}} t_j. \tag{2.14}$$

Using Equation (2.12) and Equation (2.13) for every cycle segment lead to the feature matrix $\mathbf{F}_{\mathbf{E},ALA} \in \mathbb{R}^{m\times 2s_{ALA}}$, which contains row-wise, for every cycle, the mean values in the first s_{ALA} columns, and the remaining ones contain the slopes. If $n > 500$, downsampling with the nearest integer of $\frac{n}{500} + 1$ as the factor is performed to reduce the computational cost of the ALA algorithm.

2.2.1.2 Best Daubechies Wavelets

Using a wavelet transform for signal analysis has many applications in science and engineering, cf. [60–62]. The special feature of the wavelet transform is that it offers high resolution in the frequency domain and low resolution in the time domain for low-frequency signals and vice versa for high-frequency signals [63]. A simultaneously high temporal and high spectral resolution is impossible due to the Gabor limit [64], which states that there is always a trade-off between time and frequency resolution. In comparison to the wavelet transform, the Fourier transform has a constant frequency

resolution over the entire frequency range. Thus, a wavelet transform is the better choice for extracting features in the time-frequency domain from a signal with a dynamic frequency spectrum.

The *continuous Wavelet transform* (CWT) $\mathcal{W}_\psi f(a,b)$ of a function $f(t)$ is defined by

$$\mathcal{W}_\psi f(a,b) := |a|^{-\frac{1}{2}} \int_{-\infty}^{\infty} f(t) \cdot \psi\left(\frac{t-b}{a}\right) dt, \tag{2.15}$$

where

$$\psi_{ab}(t) = |a|^{-\frac{1}{2}} \psi\left(\frac{t-b}{a}\right) \tag{2.16}$$

denotes the continuous mother wavelet, which can be scaled by a and translated by b [65]. In contrast to the CWT, the *discrete Wavelet transform* (DWT) is based on discretely sampled wavelets. A DWT can be calculated by a fast wavelet transform implemented very efficiently as a filter bank [66], i.e., as a sequence of low-pass and high-pass filters, see Figure 2.6.

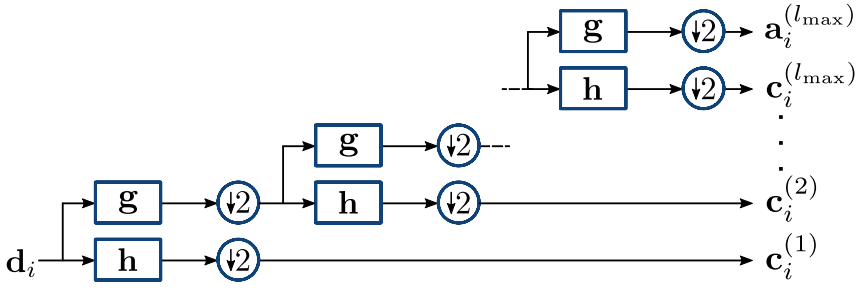


Figure 2.6: Filter bank for the DWT implementation.

Best Daubechies Wavelets (BDW) uses a DWT with a Daubechies-4 ($D4$) wavelet (four wavelet and scaling function coefficients) as the mother wavelet. The family of Daubechies wavelets was introduced by the Belgian mathematician Ingrid Daubechies in 1992 [67] and belongs to the class of orthogonal wavelets. The name DN of an individual Daubechies wavelet contains the length $N = 2p$ of the filter. Each wavelet has a maximum of p vanishing moments, i.e., it is orthogonal to every polynomial with degree $< p - 1$. So, the $D4$ wavelet, shown in Figure 2.7, has two vanishing moments and is orthogonal to all linear and constant functions.

No algebraic formula exists for the scaling function of the $D4$ wavelet, but the coefficients of the filters express it. According to [67], the four high-frequency filter

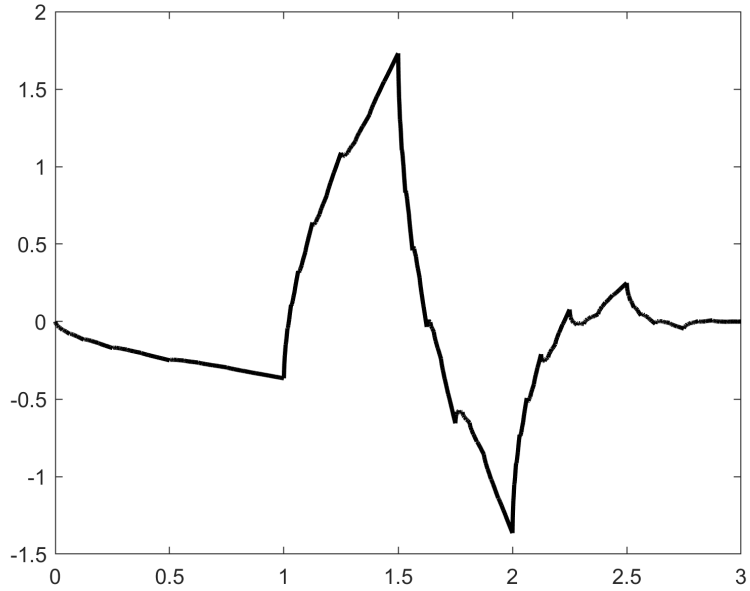


Figure 2.7: $D4$ mother wavelet (adapted from [67]).

coefficients h and the four low-frequency filter coefficients g for the $D4$ wavelet are given by

$$\mathbf{g} = (g_1, g_2, g_3, g_4) = \left(\frac{1-\sqrt{3}}{4\sqrt{2}}, \frac{3-\sqrt{3}}{4\sqrt{2}}, \frac{3+\sqrt{3}}{4\sqrt{2}}, \frac{1+\sqrt{3}}{4\sqrt{2}} \right) \tag{2.17}$$

and

$$\mathbf{h} = (h_1, h_2, h_3, h_4) = (-g_4, g_3, -g_2, g_1)\,. \tag{2.18}$$

As wavelet transforms are linear, they can be defined by matrices of dimension $n \times n$ containing the filter coefficients if the input signal is of size n. Each level $l \in \mathbb{N}_{>0}$ of the filter bank performs a parallel high-pass and low-pass filtering of the input data $\mathbf{a}_i^{(l-1)}$ followed by downsampling by 2. The wavelet transform provides a decomposition of the signal into coarse information contained in the approximation vectors $\mathbf{a}_i^{(1)}, \mathbf{a}_i^{(2)}, \ldots, \mathbf{a}_i^{(l_{\max})}$ and detailed information contained in the detail vectors $\mathbf{c}_i^{(1)}, \mathbf{c}_i^{(2)}, \ldots, \mathbf{c}_i^{(l_{\max})}$. $l_{\max}$ denotes the maximum number of wavelet decomposition levels. This number depends not only on the length of the input data but also on the chosen mother wavelet. For a wavelet transform with the $D4$ wavelet, the maximum number of wavelet decomposition levels

$l_{\max}$ for a signal of length n is determined by

$$3 \cdot 2^{l_{\max}} \leq n \Leftrightarrow l_{\max} \leq \text{lb}\left(\frac{n}{3}\right), \quad l_{\max} \in \mathbb{N}. \tag{2.19}$$

The function $\text{lb}(a)$ in the equation above denotes the logarithm to the base 2 of a, i.e., $\log_2(a)$. After $l_{\max}$ wavelet decomposition levels, the Wavelet coefficients for the i-th cycle are given by $(\mathbf{a}_i^{(l_{\max})}, \mathbf{c}_i^{(l_{\max})}, \ldots, \mathbf{c}_i^{(2)}, \mathbf{c}_i^{(1)})$. 10 % of the wavelet coefficients with the highest average absolute value over all cycles are chosen as features because they contribute the most to a low approximation error [68]. According to [37], choosing only 10 % of the wavelet coefficients is a suitable trade-off between a low approximation error and a low number of features leading to a significant feature reduction. This leads to the feature matrix $\mathbf{F}_{\mathbf{E},BDW} \in \mathbb{R}^{m \times s_{BDW}}$, which contains row-wise the s_{BDW} chosen wavelet coefficients for every cycle.

2.2.1.3 Best Fourier Coefficients

Best Fourier Coefficients (BFC) performs a *discrete Fourier transform* (DFT) to extract features from the frequency domain. The DFT for a real-valued signal $\mathbf{x}$ of length n, i.e., $\mathbf{x} = (x_0, x_1, \ldots, x_{n-1})$, is defined as

$$\begin{aligned}
\mathcal{X}_k &= \sum_{l=0}^{n-1} x_l \exp\left(-jk\beta_l\right) && (2.20)\\
&\overset{\text{Euler}}{=} \sum_{l=0}^{n-1} x_l \cdot \left[\cos\left(k\beta_l\right) - j \cdot \sin\left(k\beta_l\right)\right] && (2.21)\\
&= \sum_{l=0}^{n-1} x_l \cdot \cos\left(k\beta_l\right) - j \cdot \sum_{l=0}^{n-1} x_l \cdot \sin\left(k\beta_l\right) && (2.22)\\
&= \Re_k + j \cdot \Im_k, \qquad k = 0, \ldots, n-1 && (2.23)
\end{aligned}$$

with $j = \sqrt{-1}$ and $\beta_l = 2\pi\frac{l}{n}$ [69, 70]. As the DFT of real-valued signals is symmetric, i.e., $\mathcal{X}_k = \mathcal{X}_{n-k}^*$, only $\frac{n}{2} + 1$ Fourier coefficients must be computed. In the AMLT, amplitude and phase representation of the DFT are used, i.e.,

$$A_k = \sqrt{\Re_k^2 + \Im_k^2} \qquad \text{and} \qquad P_k = \arctan\left(\frac{\Im_k}{\Re_k}\right), \tag{2.24}$$

which are non-linear transformations in contrast to real and imaginary part representations.

For every cycle $\mathbf{d}_i$ of the data matrix $\mathbf{D}$, the amplitudes A_{ik} and the phases P_{ik} are calculated, and 10 % of the amplitudes with the highest average absolute value over all cycles and their corresponding phases are extracted as features from the frequency domain [68]. As for BDW, here again, 10 % is a suitable trade-off between a low approximation error and a low number of features [37]. This leads to the feature matrix $\mathbf{F}_{\mathbf{E},BFC} \in \mathbb{R}^{m \times 2s_{BFC}}$, which contains row-wise the $s_{BFC} = \frac{n}{20}$ chosen amplitudes and their corresponding phases per cycle.

2.2.1.4 Principal Component Analysis

To extract information contained in the general cycle shape, *Principal Component Analysis* (PCA) can be used [37]. PCA is a linear transformation that reduces the dimensionality of a data set by transforming the data set to a new coordinate system while preserving as much information as possible in as few new variables, the so-called principal components (PCs), as possible [71–74]. The PCs, i.e., the eigenvectors of the covariance matrix of the data set, represent the axes directions of the new PC space while the corresponding eigenvalues explain the variance in the data along the new axes. PCs are orthogonal and, therefore, uncorrelated. The first PC is the direction with the largest variance, the second PC is orthogonal to the first and explains the second largest variance, and so on for subsequent PCs. For dimensionality reduction, only the first r PCs are kept. Thus, for the data matrix $\mathbf{D} \in \mathbb{R}^{m \times n}$, it holds

$$\mathbf{D} = \mathbf{T} \cdot \mathbf{W}^{\top}, \tag{2.25}$$

where $\mathbf{T} \in \mathbb{R}^{m \times r}$, $r < m$, denotes the scores matrix consisting row-wise of the projections of the corresponding vector of $\mathbf{D}$ onto the eigenvectors contained column-wise in the coefficient matrix $\mathbf{W} \in \mathbb{R}^{n \times r}$. In MATLAB®, the decomposition of $\mathbf{D}$ is carried out by a *singular value decomposition* (SVD) [75]. Figure 2.8 presents an example of a PCA performed on a two-dimensional data set consisting of random data.

For this data set (cf. Figure 2.8(a)), the new axes represented by the two eigenvectors are shown in red. The green ellipse in this plot is the confidence ellipse that contains 95.45 % (2σ range) of the data. The data set transformed into the new space is shown in Figure 2.8(b).

If $n > 500$, downsampling with the nearest integer of $\frac{n}{500} + 1$ as the factor is performed. The first principal components' projections are used as features from the time domain.

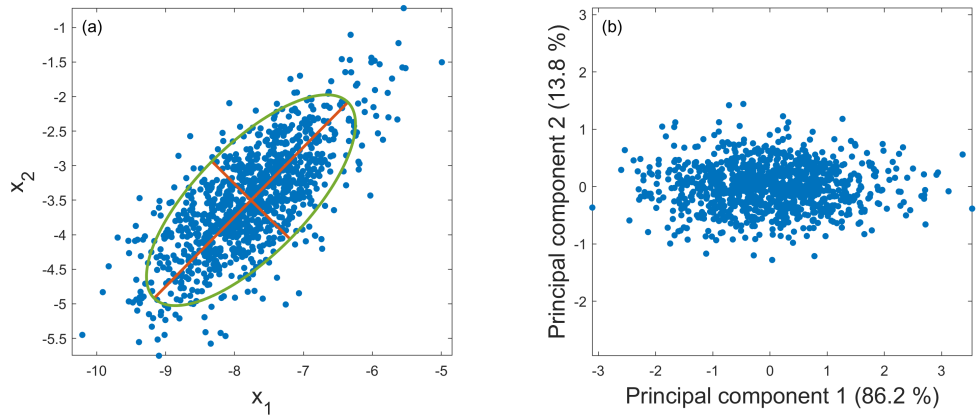


Figure 2.8: Principal Component Analysis performed on a two-dimensional data set consisting of random data. (a) Axis of the new feature space represented by the eigenvectors (red) and confidence ellipse of the 2σ range (green). (b) Data in the new space.

This leads to the feature matrix $\mathbf{F}_{\mathbf{E},PCA} \in \mathbb{R}^{m \times s_{PCA}}$, which contains $s_{PCA} \leq 500$ principal components.

2.2.1.5 Statistical Moments

Statistical Moments (SM) describe the characteristics of the statistical distribution of measurement values, which also contain information in the time domain [76]. Therefore, the first four statistical moments (mean, standard deviation as the root of the variance, skewness, and kurtosis) are used to extract features. Instead of calculating the statistical moments over complete cycles, which would lead to only four features per cycle, the cycles are each divided into $s_{SM} = 10$ segments of nearly equal length, and the statistical moments are calculated for each of the segments to extract ten times more features. The division into nearly equally sized segments and extracting statistical moments as features were already successfully performed by Helwig et al. [77].

Let $\mathbf{a} = (a_1, a_2, \ldots, a_{s_{SM}})$ and $\mathbf{e} = (e_1, e_2, \ldots, e_{s_{SM}})$ denote the indices of the first and the last measurement value of every segment, respectively. The indices can be calculated according to

$$a_k = (k-1) \cdot \left\lceil \frac{n}{s_{SM}} \right\rceil + 1 \qquad \text{and} \tag{2.26}$$

$$e_k = \min\left(k \cdot \left\lceil \frac{n}{s_{SM}} \right\rceil, n\right) \tag{2.27}$$

for $k = 1, \ldots, s_{SM}$. It holds $a_1 = 1$ and $e_{s_{SM}} = n$. The number of measurement values of the k-th segment is given by $N_k = e_k - a_k + 1$ for $k = 1, \ldots, s_{SM}$, where N_k is equal for the same segment for each cycle. The four statistical moments of the k-th segment of cycle $\mathbf{d}_i$ can be calculated as follows [78]:

- mean value

$$\mu_{ik} = \overline{\mathbf{d}_{ik}} = \frac{1}{N_k} \sum_{j=a_k}^{e_k} d_{ij}, \tag{2.28}$$

- standard deviation

$$\sigma_{ik} = \sqrt{\frac{1}{N_k - 1} \sum_{j=a_k}^{e_k} (d_{ij} - \mu_{ik})^2}, \tag{2.29}$$

- skewness

$$v_{ik} = \frac{\frac{1}{N_k} \sum\limits_{j=a_k}^{e_k} (d_{ij} - \mu_{ik})^3}{\left(\frac{1}{N_k} \sum\limits_{j=a_k}^{e_k} (d_{ij} - \mu_{ik})^2 \right)^{\frac{3}{2}}}, \qquad \text{and} \tag{2.30}$$

- kurtosis

$$w_{ik} = \frac{\frac{1}{N_k} \sum\limits_{j=a_k}^{e_k} (d_{ij} - \mu_{ik})^4}{\left(\frac{1}{N_k} \sum\limits_{j=a_k}^{e_k} (d_{ij} - \mu_{ik})^2 \right)^2}. \tag{2.31}$$

The resulting feature matrix $\mathbf{F}_{\mathbf{E},SM} \in \mathbb{R}^{m \times 4s_{SM}}$ contains, row-wise per cycle, the four statistical moments for each of the s_{SM} sections.

2.2.2 Feature Selection

For Big Data applications, the dimensionality reduction will be insufficient after FE. Therefore, further dimensionality reduction is performed by selecting only the most relevant features concerning the given task at hand. An ideal feature set contains a small number of uncorrelated features with a high variance. As the target value for every cycle is known, FS is a supervised dimensionality reduction step. Unlike FE, which

generates new features from the data, FS keeps a subset of the existing feature set $\mathbf{F_E}$ by removing redundant, irrelevant, and noisy features from the feature set. Therefore, FS enables faster model training, reduces the complexity and improves the accuracy of the ML model, and reduces overfitting [79, 80]. Overfitting means the model has learned too much from the data and starts memorizing it instead of understanding it [81], leading to performance degradation on data that was not used for training [79].

In the AMLT, three complementary algorithms, chosen according to [57], are used to rank features and filter redundant, irrelevant, and noisy features from the feature matrices $\mathbf{F_E}$. In general, FS algorithms can be divided into filter, wrapper, and embedded methods. Filter methods, e.g., Pearson correlation (cf. Section 2.2.2.1) and ReliefF (cf. Section 2.2.2.2), select features according to statistical techniques which indicate the relationship between the feature and the corresponding target without training a model. Wrapper methods, e.g., *Recursive Feature Elimination Support Vector Machine* (RFESVM) (cf. Section 2.2.2.3), have high computational cost as these methods repeatedly train a model on a feature subset, determine the performance of the trained model, change the feature subset by adding or removing features, train a new model, and compare the performance of both models. Thus, the model detects the importance of the features by learning. Embedded methods, which are not used in the AMLT, combine the qualities of wrapper and filter methods, i.e., the selection process is embedded in the learning. In contrast to wrapper methods, only one ML model is trained, which leads to less computational cost, and features are selected based on their importance returned by this trained model. Examples of embedded methods are *Least Absolute Shrinkage and Selection Operator* (LASSO) [29, 82], ridge [83, 84], and elastic net [85].

To determine the optimum number of features, a 10-fold CV, explained in more detail in Section 2.2.4, is carried out for every number of features in each feature subset using a suitable ML algorithm, as explained in Section 2.2.3. The lowest CV error, according to this brute force approach, leads to the optimum number l of the most relevant features. Thus, FE can be mathematically defined as a mapping $\mathbf{F_E} \mapsto \mathbf{F_S}$, where $\mathbf{F_E} \in \mathbb{R}^{m \times k}$, $k \in \mathbb{N}$, denotes the matrix containing extracted features and $\mathbf{F_S} \in \mathbb{R}^{m \times l}$, $l \leq k$, contains the optimum number of most relevant features.

2.2.2.1 Pearson Correlation

Pearson correlation is a filter method used for feature preselection as well as selection in the AMLT. If the number of features is more than 500 per feature set $\mathbf{F_E}$, a preselection is performed by Pearson correlation due to its low computational cost. In general, Pearson

correlation measures the strength and, if it is not the absolute value, the direction of the linear relationship between a feature and a target in terms of the Pearson correlation coefficient $r_{Pearson} \in [-1, 1]$. If $r_{Pearson}$ is close to zero, no linear relationship is indicated. The Pearson correlation coefficient $r_{Pearson,j}$ for a feature, i.e., one column $\mathbf{f}_j$ of $\mathbf{F_E}$, and a target $\mathbf{y}$ is defined as

$$r_{Pearson,j} = \frac{\sum_{i=1}^{m} \left((f_{ij} - \overline{\mathbf{f}_j})(y_i - \overline{\mathbf{y}}) \right)}{\left[\sum_{i=1}^{m} \left((\mathbf{f}_{ij} - \overline{\mathbf{f}_j})^2 \right) \sum_{i=1}^{m} \left((y_i - \overline{\mathbf{y}})^2 \right) \right]^{1/2}}, \tag{2.32}$$

where m denotes the number of cycles [86, 87]. The features in the AMLT are sorted in descending order according to their Pearson correlation coefficient, and the first $l_{Pearson}$ features for all m cycles build the matrix $\mathbf{F}_{\mathbf{S},Pearson} \in \mathbb{R}^{m \times l_{Pearson}}$.

2.2.2.2 ReliefF

Like Pearson correlation, ReliefF belongs to the filter methods and is used in case of classification tasks when no linear class separation is possible. As the version naming of this algorithm is according to the Latin alphabet, the F denotes the sixth version of the algorithm Relief, which is, in its basic version, limited to binary classification problems [88, 89]. In contrast to Relief, ReliefF can deal with multi-class problems. This algorithm determines the nearest hits and the nearest misses for each point by using k-nearest neighbors with the Manhattan metric, which is induced by 1-norm as distance measure [90–93]. In other words, for one point belonging to one class, ReliefF searches for k of the nearest neighbors from the same class (nearest hits) and k of the nearest neighbors from each different class (nearest misses) [56]. In the AMLT, k is set to three. The contribution of all nearest hits and all nearest misses are averaged to update the quality estimations of the features. These weights indicate the ranking of the features, which means that $\mathbf{F}_{\mathbf{S},ReliefF} \in \mathbb{R}^{m \times l_{ReliefF}}$ contains the $l_{ReliefF}$ features with the highest weights for all m cycles.

2.2.2.3 Recursive Feature Elimination Support Vector Machine

Recursive Feature Elimination Support Vector Machine (RFESVM) is a wrapper method introduced in [94, 95]. This method has higher computational cost and a higher risk of overfitting than filter methods [96, 97]. Using One-vs-One classification, multi-class problems are split into several binary classification problems by setting one class as positive, another as negative, and ignoring all the other classes. This means that for N_c

classes, all pairs of classes are used, and $\binom{N_c}{2}$ binary classification problems are solved [98]. In the simplest case of a binary classification problem, Support Vector Machines (SVMs) find an optimal hyperplane

$$\mathbf{w}^\top \mathbf{x} + b = 0, \tag{2.33}$$

which separates the two classes by determining the maximum margin $\frac{2}{\|\mathbf{w}\|}$, i.e., the maximum distance between the hyperplane and the support vectors of both classes. $\mathbf{w}$ is the weight vector, and scalar b is called bias. The corresponding optimization problem to solve is [99]:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to} \quad & y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq 1, \quad i = 1, \ldots, l. \end{aligned} \tag{2.34}$$

In this equation, $\mathbf{x}_i$ are the support vectors, and $y_i = \pm 1$ are the labels. Figure 2.9a shows a linear hard-margin SVM for binary classification in the two-dimensional space. The support vectors for the positive and the negative class are labeled with surrounding green circles. In the two-dimensional case, the hyperplanes are lines. Support vectors of a hard-margin SVM lie on these lines. A hard-margin SVM does not tolerate outliers and, therefore, does not work if the data is not linearly separable.

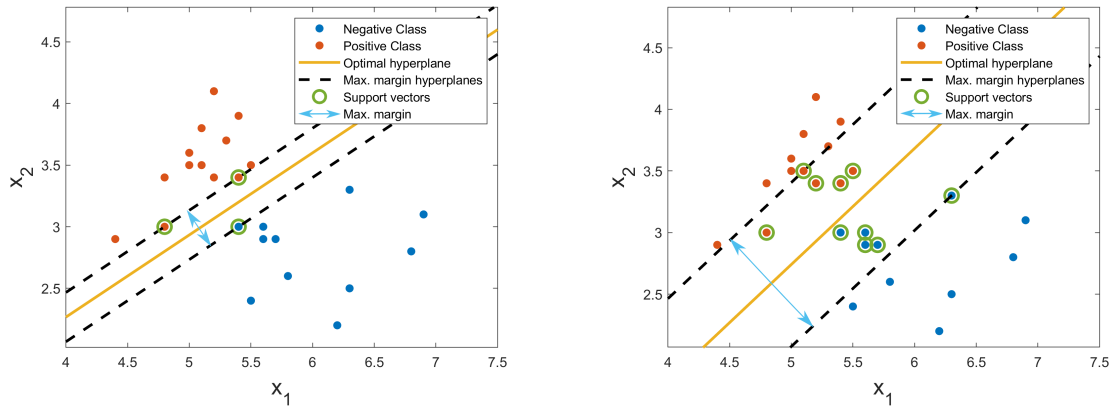
In contrast to a hard-margin SVM, a soft-margin SVM allows data separation with outliers. In the AMLT, a linear soft-margin SVM, i.e., a soft-margin SVM with a linear kernel, is implemented to solve the following optimization problem [29]:

$$\begin{aligned} \min_{\mathbf{w},b,\xi_i} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l} \zeta_i \\ \text{subject to} \quad & y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq 1 - \zeta_i, \quad \zeta_i \geq 0, \quad i = 1, \ldots, l. \end{aligned} \tag{2.35}$$

In this equation, $\mathbf{x}_i$ denotes the support vectors, $y_i = \pm 1$ the labels, C the regularization parameter, and ζ_i the slack variables. The smaller the regularization parameter C is, the wider the margin. A smaller regularization parameter C increases the importance of the slack variables ζ_i, whereas a higher C decreases the importance of the ζ_i. The hard-margin SVM (cf. Figure 2.9a) corresponds to $C = \infty$.

In Figure 2.9b, a linear soft-margin SVM for binary classification is shown in the two-dimensional space. The support vectors for the positive and the negative class are

labeled with surrounding green circles. In the two-dimensional case, the hyperplanes are lines. A soft-margin SVM allows support vectors to lie within the margin.



(a) Linear hard-margin Support Vector Machine. (b) Linear soft-margin Support Vector Machine.

Figure 2.9: Comparison of (a) linear hard-margin and (b) linear soft-margin Support Vector Machine for binary classification in the two-dimensional space.

For multi-class problems, the weights are added up over all binary classification problems. Features with the lowest weights are recursively removed from the feature set $\mathbf{F_E}$ as they contribute the least to the class separation [100]. Thus, $\mathbf{F_S}_{,RFESVM} \in \mathbb{R}^{m \times l_{RFESVM}}$ contains the l_{RFESVM} most relevant features according to the SVM weights for all m cycles.

2.2.3 Machine learning algorithms

FE and FS, introduced in the previous sections, prepare the data for using ML algorithms. This means that for each combination of FE and FS, a matrix $\mathbf{F_S} \in \mathbb{R}^{m \times l}$ containing the optimum number l of the most relevant features is given as input for the last step of the AMLT. In general, ML algorithms are divided into supervised, unsupervised, semi-supervised, and reinforcement techniques (cf. Figure 2.4)). Supervised learning techniques analyze labeled data sets, whereas unsupervised learning techniques use unlabeled data sets. Classification and regression are supervised machine learning techniques. Clustering methods belong to unsupervised ML algorithms and try to group data [51, 101, 102]. Reinforcement learning algorithms, e.g., Q-learning [103, 104], learn from experiences in a feedback-based process. An overview of reinforcement algorithms is given in [105]. Most literature distinguishes only between these three kinds of machine learning techniques. Some newer literature, such as [46], adds semi-supervised learning

as a fourth kind of ML technique. While (un)supervised learning means that the data set is (un)labeled, semi-supervised learning uses labeled as well as unlabeled data. Novelty detection, which identifies new or unknown data, can be named as a representative of this learning technique [106, 107].

In the AMLT, only supervised machine learning algorithms are used. As shown in Figure 2.5, a distinction is made between the toolbox for classification problems and the toolbox for regression problems. The objective of classification or quantification is to apply a trained model on unknown observations (cycles) and predict their class labels, e.g., failure status for condition monitoring and predictive maintenance, or their individual response values, e.g., gas concentrations to determine the indoor air quality (IAQ). Data sets for these use cases are introduced in Section 2.3.

2.2.3.1 Classification

In the AMLT, the classification is divided into two parts. First, further dimensionality reduction is performed using LDA in order to reduce computational cost and avoid overfitting. In addition to overfitting, the curse of dimensionality, also called Hughes phenomenon [108], can lead to models with lower accuracy due to counterintuitive geometrical properties of high dimensional spaces. Thus, the curse of dimensionality must also be avoided. The goodness of the classification is expressed by the classification error, defined as the percentage of misclassified cycles.

LDA is a linear dimensionality reduction technique first introduced by Fisher in 1936 using the well-known multivariate Fisher's iris data set as an example [109]. The objective of LDA is to minimize the within-class (intra-class) variance and maximize the between-class (inter-class) variance [110]. In contrast to PCA, which finds the axes maximizing the variance within the data, as explained in Section 2.2.1.4, LDA maximizes the axes for class separability. The within-class and the between-class scatter matrix for classes C_i, $i = 1, \ldots, n_{class}$, are given by

$$\mathbf{S_W} = \sum_{i=1}^{n_{class}} \sum_{\mathbf{x} \in C_i} (\mathbf{x} - \overline{\mathbf{x}_i}) (\mathbf{x} - \overline{\mathbf{x}_i})^t \tag{2.36}$$

and

$$\mathbf{S_B} = \sum_{i=1}^{n_{class}} N_i (\overline{\mathbf{x}_i} - \overline{\mathbf{x}}) (\overline{\mathbf{x}_i} - \overline{\mathbf{x}})^t , \tag{2.37}$$

respectively [56, 111, 112]. The number of observations per class C_i is given by the scalar N_i, and the vector containing features by $\mathbf{x}$. $\overline{\mathbf{x}_i}$ denotes the column-wise arithmetic mean vector of the features belonging to class C_i, and $\overline{\mathbf{x}}$ is the grand mean for the whole feature set. The objective of the LDA is to obtain the optimal projection matrix $\mathbf{W}$ by maximizing Fisher's criterion [56]

$$J(\mathbf{W}) = \frac{\mathbf{W}^\top \mathbf{S_B} \mathbf{W}}{\mathbf{W}^\top \mathbf{S_W} \mathbf{W}}. \tag{2.38}$$

The optimal projection matrix $\mathbf{W}$ can be achieved by solving the generalized eigenproblem

$$\mathbf{S_B}\mathbf{w}_i = \lambda_i \mathbf{S_W}\mathbf{w}_i \quad \Leftrightarrow \quad \mathbf{S_W^{-1}}\mathbf{S_B}\mathbf{w}_i = \lambda_i \mathbf{w}_i, \tag{2.39}$$

where $\mathbf{w}_i$ denotes the i-th eigenvector corresponding to the i-th largest eigenvalue λ_i [113]. For n_{class} classes, LDA performs a linear projection of the feature space into an $(n_{class} - 1)$-dimensional subspace, i.e., $\mathbf{W} \in \mathbb{R}^{l \times (n_{class}-1)}$. It holds

$$\mathbf{F}_{LDA} = \mathbf{F_S} \cdot \mathbf{W} \tag{2.40}$$

with $\mathbf{F}_{LDA} \in \mathbb{R}^{m \times (n_{class}-1)}$. It should be noted that LDA assumes a normal distribution of the data within each class, which is usually not the case for real-world data. As LDA is quite robust against violating the assumption of a normal distribution, it can also be applied to classes whose data are not normally distributed [110, 114].

For the example in Figure 2.10, the multivariate Fisher's iris data set [109] is used, which contains 50 samples for each of the three iris species (setosa, versicolor, virginica) with four features (length and width of sepals and petals). The data for the sepal measurements is presented in Figure 2.10(a). The LDA plot (cf. Figure 2.10(b)) shows that the first discriminant function separates the data well, whereas the second discriminant function does not contribute much to the class separation.

After performing LDA, the actual classification is carried out using the Mahalanobis distance [115–117]. It measures distances relative to the central point of each class and takes correlation into account by considering the covariance matrix (cf. Equation (2.41)). Points of the same Mahalanobis distance towards the central point of a class form an ellipse around the central point [118, 119], whereas points of the same Euclidean distance build a circle in the two-dimensional space.

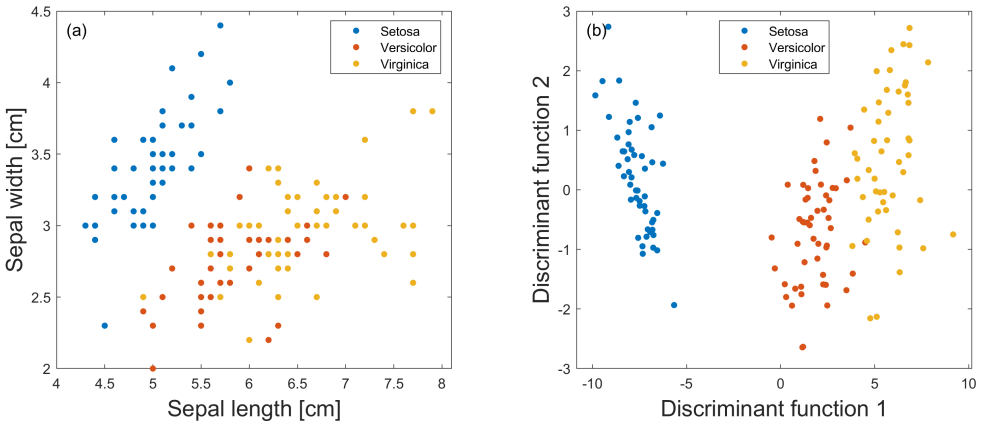


Figure 2.10: Linear Discriminant Analysis on the multivariate Fisher's iris data set. (a) Sepal measurement data. (b) Projection of the Fisher's iris data into the new two-dimensional subspace.

The Mahalanobis distance between $\mathbf{f}$ and the central point of class C_i is defined as

$$d_{Mahal}\left(\mathbf{f}\right) = \sqrt{\left(\mathbf{f} - \overline{\mathbf{x}_i}\right)^{\top} \mathbf{S}_i^{-1} \left(\mathbf{f} - \overline{\mathbf{x}_i}\right)}. \tag{2.41}$$

In this equation, $\mathbf{f}$ is the vector of the test data features, $\overline{\mathbf{x}_i}$ is the column-wise arithmetic mean of the training data features belonging to class C_i, and $\mathbf{S}_i$ is the covariance matrix of class C_i. The class with the lowest Mahalanobis distance $d_{Mahal} \in \mathbb{R}_{\geq 0}$ is assigned to $\mathbf{f}$. In the AMLT, Mahalanobis distance classification, together with a previously performed LDA, is chosen to determine the corresponding class for an unknown point as it is less computational cost-intensive than *K-nearest neighbors* (KNN) classification, which assigns unknown points to the class that is most represented within the K nearest neighbors [51].

2.2.3.2 Regression

In the AMLT, *Partial Least Squares Regression* (PLSR) is used as the quantification algorithm. The objective of PLSR is to model the dependence relationship between the target $\mathbf{y} \in \mathbb{R}^m$ and multiple features $\mathbf{F_S} \in \mathbb{R}^{m \times l}$ [120]. The goodness of regression is expressed by the RMSE, which is a measure of the differences between the observed $\mathbf{y} \in \mathbb{R}^m$ and the predicted target values $\hat{\mathbf{y}} \in \mathbb{R}^m$ and is defined by

$$\mathrm{RMSE}(\mathbf{y}, \hat{\mathbf{y}}) = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2}. \tag{2.42}$$

In [121], the basic PLSR algorithm of a response matrix on a predictor matrix using n_{comp} PLSR components is developed. In case of the AMLT, let the predictor matrix be given as the selected feature matrix $\mathbf{F_S} \in \mathbb{R}^{m \times l}$ and the response matrix be only a vector, i.e., $\mathbf{y} \in \mathbb{R}^m$. To perform PLSR, the following decompositions must be iteratively solved such that the covariance between $\mathbf{F_S}$ and $\mathbf{y}$ is maximized [29, 122]:

$$\mathbf{F_S} = \widehat{\mathbf{F_S}} + \mathbf{F}_{\mathbf{S}_{\text{res}}} = \mathbf{F}_{\mathbf{S}_S} \cdot \mathbf{F}_{\mathbf{S}_L}^{\top} + \mathbf{F}_{\mathbf{S}_{\text{res}}} \qquad \text{and} \tag{2.43}$$

$$\mathbf{y} = \hat{\mathbf{y}} + \mathbf{y}_{\text{res}} = \mathbf{Y}_S \cdot \mathbf{y}_L^{\top} + \mathbf{y}_{\text{res}}, \tag{2.44}$$

where $\mathbf{F}_{\mathbf{S}_L} \in \mathbb{R}^{l \times n_{\text{comp}}}$ and $\mathbf{y}_L \in \mathbb{R}^{1 \times n_{\text{comp}}}$ denote the loading matrix and loading vector, respectively. $\mathbf{F}_{\mathbf{S}_S} \in \mathbb{R}^{m \times n_{\text{comp}}}$ and $\mathbf{Y}_S \in \mathbb{R}^{m \times n_{\text{comp}}}$ are the predictor and response scores, respectively. The matrix $\mathbf{F}_{\mathbf{S}_{\text{res}}}$ and the vector $\mathbf{y}_{\text{res}}$ are the residual terms for predictor and response and are used as the start for the next iteration step. In these equations, $\widehat{\mathbf{F_S}}$ and $\hat{\mathbf{y}}$ give the *Partial Least Square* (PLS) estimations of $\mathbf{F_S}$ and $\mathbf{y}$, respectively.

An example of a PLSR model for predicting formaldehyde (CH_2O) concentrations based on the gas sensor data set introduced in Section 2.3.3 is presented in Figure 2.11. The training and the testing are performed with 80 % and 20 % of the data set, respectively.

In MATLAB®, PLSR is calculated using the *statistically inspired modification of the PLS* (SIMPLS) algorithm [123], which directly determines the regression coefficients without SVD or matrix inversion [124]. In the SIMPLS algorithm, a vector of ones is prepended to $\mathbf{F_S}$ to compute coefficient estimates for a model with constant terms. This augmented matrix is denoted by $\widetilde{\mathbf{F_S}} \in \mathbb{R}^{m \times (l+1)}$. For the SIMPLS algorithm, it holds

$$\mathbf{F}_{\mathbf{S}_S} = \widetilde{\mathbf{F_S}} \cdot \mathbf{W} \qquad \text{and} \tag{2.45}$$

$$\hat{\mathbf{y}} = \mathbf{F}_{\mathbf{S}_S} \cdot \mathbf{y}_L^{\top} \tag{2.46}$$

with $\mathbf{W} \in \mathbb{R}^{(l+1) \times n_{\text{comp}}}$ denoting a weight matrix. Substituting Equation (2.45) into Equation (2.46) leads to the predictive linear regression model

$$\hat{\mathbf{y}} = \widetilde{\mathbf{F_S}} \cdot \mathbf{W} \cdot \mathbf{y}_L^{\top} \tag{2.47}$$

$$= \widetilde{\mathbf{F_S}} \cdot \mathbf{b}, \tag{2.48}$$

where $\mathbf{b} \in \mathbb{R}^{l+1}$ denotes the regression coefficient vector containing the intercept term in the first entry and l PLSR coefficient estimates in the others.

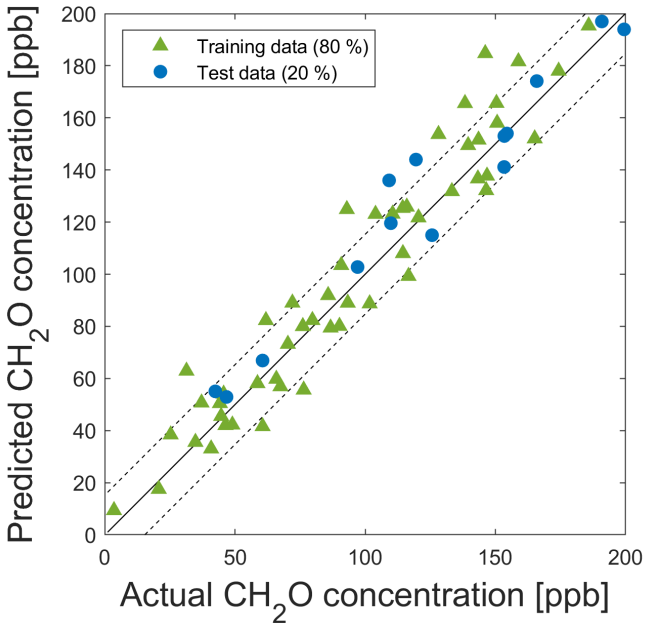


Figure 2.11: Partial Least Squares Regression model trained with training data and the concentration prediction of the training (green) and the test data (blue).

2.2.4 Validation and Testing

For evaluation of the performance of an ML model, the data set is split into three disjoint subsets: training, validation, and test data set. The training data set is used for the learning process of the model, i.e., the training, to fit parameters. With the validation data set, the model is evaluated during training, and the hyperparameters of a model, e.g., the optimum number l of the most relevant features, are tuned. This approach of splitting the data set during training prevents the trained model from overfitting, i.e., the model performs well on data already seen but cannot generalize it on previously unseen data. After completing the model training process, the final model is evaluated with the test data set, which was not seen by the trained model before.

There is no optimal split percentage and, therefore, no clear guidance on what ratio to use for the data set split. The ratio strongly depends on the use case, the total number of samples in the data set, and the hyperparameters that should be tuned. Many approaches exist in the literature, e.g., [125–128]. For the training (including validation) and the testing process, a typical split is 80:20. This split is based on the well-known Pareto principle [129], which says that 80 % of effects arise from 20 % of

causes in most cases. Thus, a commonly used split ratio is 70:10:20, which means 70 % of the data set is for training, 10 % for validation, and 20 % for testing.

The overall process of model training, validation, and testing, as well as the data set splits used in the AMLT, are shown in Figure 2.12.

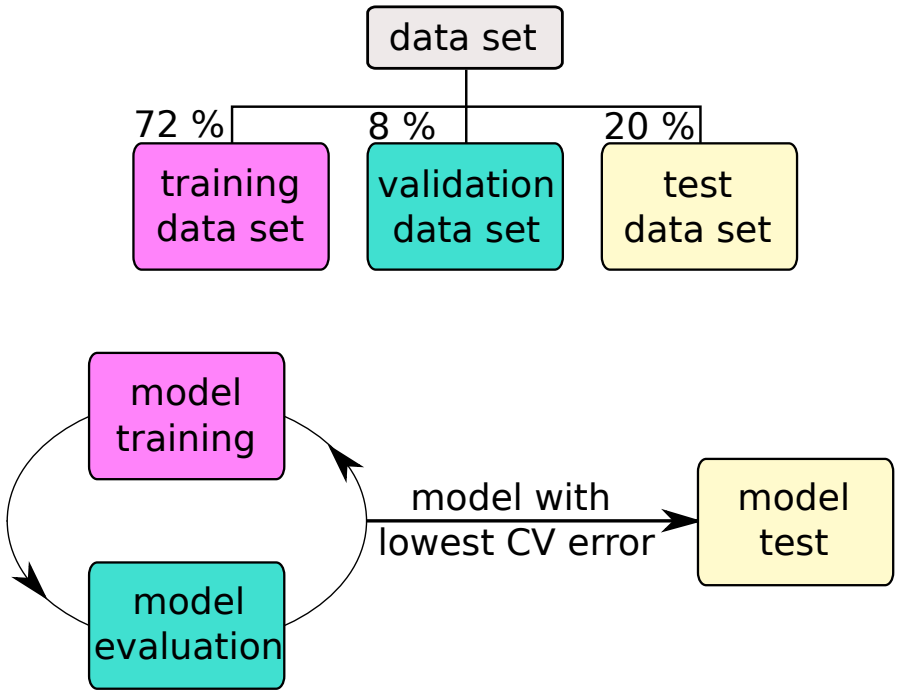


Figure 2.12: Overall process of training, validation, and testing of a model with the corresponding split of the data set used in the AMLT.

For validation in the AMLT, a k-fold stratified CV [28, 29] is automatically performed during the model training. On the one hand, if the bias is high, an ML model tends to underfit, i.e., the model is too simple and performs poorly on both training and test data. On the other hand, if the variance is high, an ML model only performs well on training data and not on test data, i.e., the model is overfitted. Models with a high variance have a low bias [130], and vice versa. This balancing of under- and overfitting is called the bias-variance trade-off [131]. Figure 2.13 illustrates the bias-variance trade-off.

Choosing a value for k depends on the data set [133]. A good compromise is given by choosing $k = 5$ or $k = 10$ [28, 29]. The commonly used split ratio 70:10:20 would be represented by $k = 8$, but in the AMLT, $k = 10$ is used, and therefore, the chosen split ratio is 72:8:20 (cf. Figure 2.12). The training data set is partitioned into k subsets

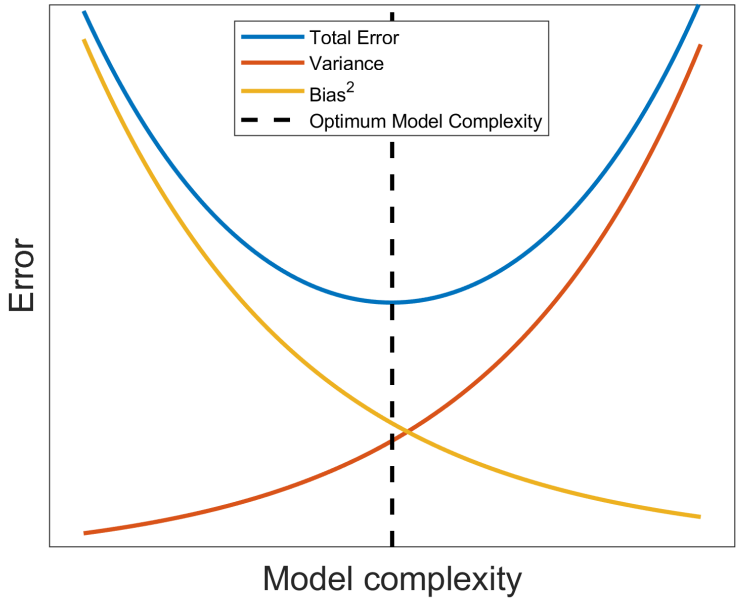


Figure 2.13: Illustration of the bias-variance trade-off (adapted from [132]).

of roughly equal size under the constraint that each subset has nearly the same label distribution as the data set itself. Model training is performed with only $k-1$ subsets, i.e., the training data. For validation, the trained model is applied to the remaining subset, i.e., the validation data, and the CV error is calculated as the performance measure during the training process. To determine the CV error, the corresponding error measure (depending on the target) is used to calculate the error for every fold, and all calculated errors are averaged over all folds [56]. As the error measure, the classification error for the classification version of the AMLT and the RMSE for the regression version is applied. The algorithm combination with the lowest CV error is chosen as the best choice for FE and FS for the task at hand.

To measure the performance of the final model, the final model is applied to the test data set, and the corresponding error, as an unbiased evaluation, is determined.

2.3 Used data sets

This section presents three different data sets, which are used to investigate the influence of measurement uncertainty in Chapter 3. As a representative for regression problems, IAQ monitoring is considered, while industrial condition monitoring is used for classifi-

cation problems. All used data sets and corresponding short descriptions are publicly available on the online service Zenodo [134].

2.3.1 Hydraulic system data set

Sustainability is a challenge in today's industry and plays a significant role, especially in Industry 4.0, with high requirements for machine availability, safety, worker health, and reliability. Machine downtime due to planned periodic maintenance or unplanned breakdown with a failure must be reduced as they are costly [135]. Therefore, condition monitoring as a key element of predictive maintenance is used to schedule optimum maintenance, avoid downtime and save money [136]. Conditions that could shorten the typical lifespan of, for example, a component or a machine are monitored and can be approached before they develop into a major problem.

Hydraulic systems are widely used in industrial applications, e.g., in extruding, forming, and punching processes, as they can deal with high forces. They use incompressible fluids, leading to more position precision and more movement control compared to pneumatic systems [137]. A simple hydraulic system to move actuators consists of a reservoir with a filter unit, e.g., an oil tank, a hydraulic pump, several valves and pipes, and a power source, which is, in many use cases, an electrical motor. In addition, accumulators are used in hydraulic systems to absorb shocks and pulsations from pressure and volume flow fluctuations. To ensure a reliable and safe hydraulic system operation, oil cooling is essential to prevent the used fluid from overheating.

The hydraulic system data set [138] contains 1,449 cycle measurements of simulated fault conditions of various components in a hydraulic system. The used hydraulic system developed in [139] consists of a primary working circuit and a secondary cooling-filtration circuit, both connected via an oil tank (cf. Figure 2.14).

It is equipped with 14 process sensors, which are already used for process control and regulation, measuring

- electrical motor power of MP1 (EPS1) with a sampling rate of 100 Hz,
- pressures (PS1 - PS6) with a sampling rate of 100 Hz,
- volume flows (FS1 and FS2) with a sampling rate of 10 Hz,
- temperatures (TS1 – TS4) with a sampling rate of 1 Hz, and
- the root-mean-square (RMS) value of the vibration velocity (VS1) with a 1 Hz sampling rate.

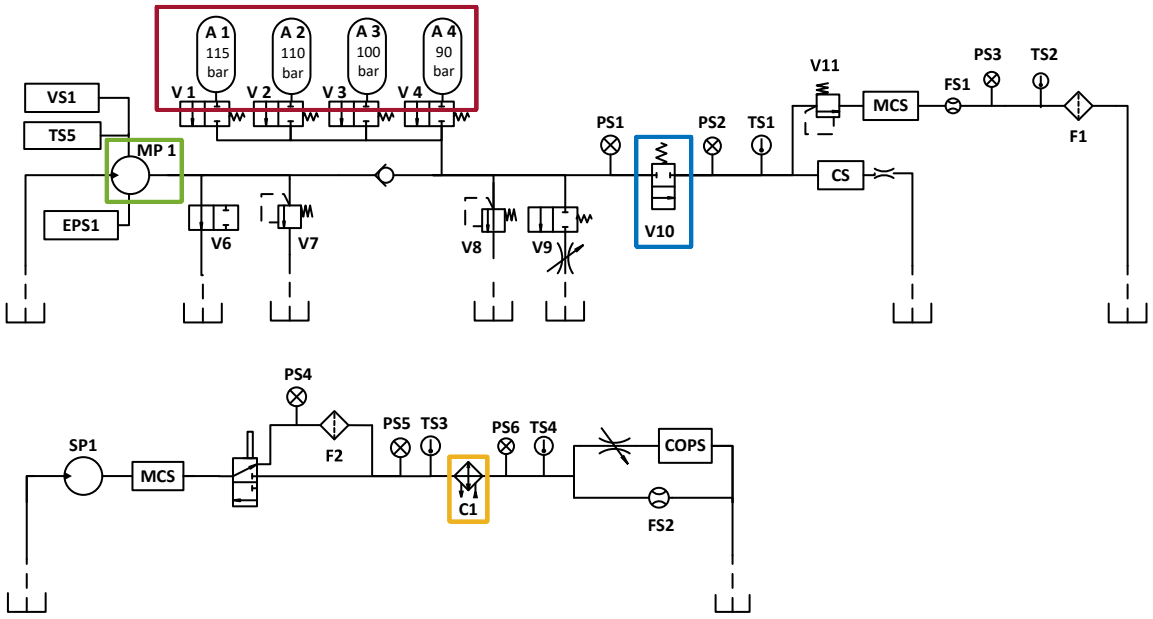


Figure 2.14: Hydraulic system, in which various fault conditions of hydraulic accumulators A1 - A4 (red), cooler C1 (yellow), pump MP1 (green), and valve V10 (blue) are simulated (adapted from [140]).

This means that no additional sensors need to be installed in addition to the existing process sensors to generate the data set for condition monitoring of the hydraulic system. Furthermore, three virtual sensors derived from data from the process sensors are included in the data set: cooling efficiency, cooling power, and system efficiency, each with a sampling rate of 1 Hz [139]. In total, measurement data of 17 sensors in SI units with different sampling rates is included in the hydraulic system data set.

In both circuits, various fault conditions of cooler, hydraulic accumulator, pump, and valve are simulated at various severity levels. An overview of these conditions, their classification target values in the data set, and the corresponding interpretations are given in Table 2.1.

Each working cycle has a constant duration of 60 s. The various fault conditions are systematically combined and changed after every tenth cycle, as shown in Figure 2.15, i.e., the data set contains ten working cycles per each of the 144 fault condition combinations. These 144 fault conditions result from the full factorial experiment design.

Each of the four fault conditions represents a continuous process, e.g., a decrease of gas filling pressure or cooling efficiency degradation, so that the data set can be used for regression. However, a classification problem is considered in this thesis as it has the benefit of additional validation besides k-fold CV by using LDA. Thereby, the

Table 2.1: Classification target values of the four different fault conditions and their interpretations (adapted from [139]). Target values are taken from the published data set [138].

Component	**Condition**	**Classes**	**Interpretation**
Accumulator A1-A4	Gas filling pressure	2	optimal pressure (130 bar)
		3	slightly reduced pressure (115 bar)
		4	severely reduced pressure (100 bar)
		5	close to total failure (90 bar)
Cooler C1	Cooling efficiency	3 %	close to total failure
		20 %	reduced efficiency
		100 %	full efficiency
Pump MP1	Internal leakage	0	no leakage
		1	weak leakage (3 × 0.2 mm)
		2	severe leakage (3 × 0.25 mm)
Valve V10	Switching behavior	73 %	close to total failure
		80 %	large delay
		90 %	small delay
		100 %	optimal switching behavior

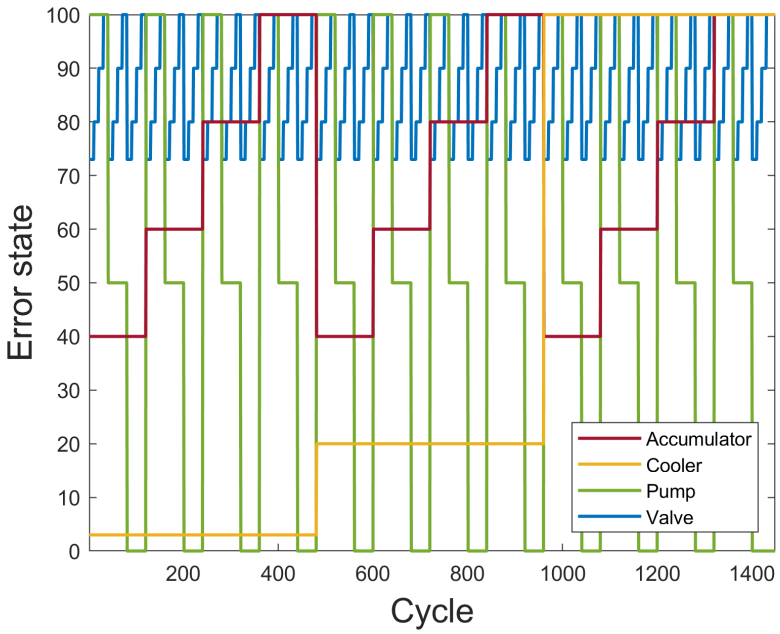


Figure 2.15: Fault conditions of hydraulic accumulators A1 - A4 (red, scaled by 20 for better visibility), cooler C1 (yellow), pump MP1 (green, scaled by 50 for better visibility), and valve V10 (blue) for 1,449 cycles.

target values are considered as discrete steps. In an LDA scatter plot, the classes are sorted ascending or descending according to their target classes, and every subsequently projected class sorts itself correctly at the position that would be expected, considering a continuous process. In contrast, if the classes are not sorted correctly, the class separation is caused not significantly by the target but by cross-influences [141]. To perform this validation using LDA, the data set is divided into training and test data. The training data is used for the model training, whereas the test data is only projected onto the LDA space. Figure 2.16 shows an example of this sorting behavior using the gas filling pressure of the accumulator as the target. The model is trained with only three classes (class 2, 3, and 5), and the unseen data (class 4) is projected onto the LDA space. The classes are sorted descending from high to low gas filling pressure.

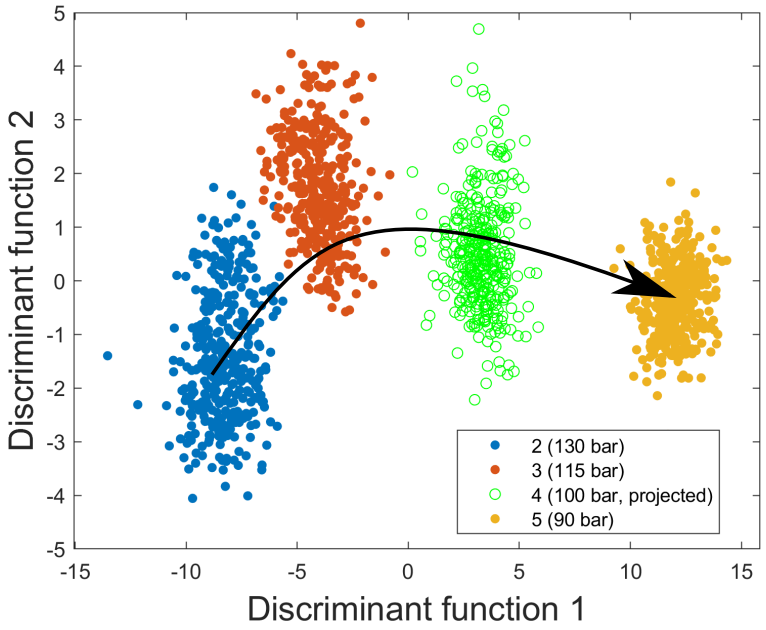


Figure 2.16: LDA plot of the gas filling pressure of the accumulator. Classes 2, 3, and 5 are used for the model training, whereas class 4 is correctly projected onto the LDA space.

2.3.2 Electromechanical cylinder data set

The electromechanical cylinder (EMC) data set [142] consists of lifetime tests of three EMCs (Festo ESBF-BS-63-400-5P [143]). It has been recorded at *Zentrum für Mechatronik und Automatisierungstechnik gGmbH* (ZeMA) with a test bed specially designed for condition monitoring and lifetime tests of EMCs [139]. The function of the EMC is

based on a ball screw drive as a mechanical linear actuator that translates a rotational into a linear motion. It has several favorable properties, e.g., high accuracy in position and repetition, low wear, and hence a long lifetime [144]. Thus, assembly and handling systems, as well as tool machines, are typical applications of EMCs. A failure of an EMC in such systems leads to a loss of quality and costly downtime. Therefore, knowledge about the actual wear and the remaining useful lifetime until the failure of the EMC occurs is of interest.

Simplified, the used test bed consists of an EMC as the device under test (DUT) and a pneumatic cylinder, which simulates a load on the DUT in axial direction during each working cycle. The combination of a high axial load, a high motion speed, and a high acceleration results in a fast wear progression of the EMC. The parameters of the lifetime test are shown in Table 2.2.

Table 2.2: Parameters for each of the three lifetime tests [141].

Parameter	**Value**
Velocity	200 mm/s
Axial force	7 kN (const. pulling)
Stroke range	100 mm to 350 mm
Acceleration	5,000 mm/s^2
Deceleration	5,000 mm/s^2

The drag error is used as failure criterion, meaning the lifetime test fails when a deviation larger than 30 mm between the set and the actual EMC end position arises [139]. Only small deviations usually occur during normal operation, whereas a significant deviation increase indicates a defect of the EMC [139]. The remaining useful lifetime (RUL), i.e., the period an EMC is likely to operate before it requires repair or replacement, is assumed to reduce linearly at this test bed based on [145]. As target, the used lifetime in percent, i.e., the opposite of RUL, is used for this data set. Used lifetime can be considered as a regression or a classification problem; however, in this thesis, the data set is used as classification problem because of the benefit of an additional validation as mentioned in Section 2.3.1. Thus, the used lifetime as classification target for this data set starts at 1 % and ends at 100 % when the EMC is detected as defective with discrete class percentages (1 % increments) in between.

The data acquisition unit (DAQ) of the EMC test bed (*ZeMA DAQ*) acquires data from eleven different sensors with sampling rates between 10 kHz and 1 MHz during

each cycle of the lifetime test. In detail, these sensors are described in the following listing [139]:

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

These sensors are localized at different positions schematically shown in Figure 2.17. It is assumed that the ZeMA DAQ provides equidistant samples in time, as no timestamps are sampled with this system.

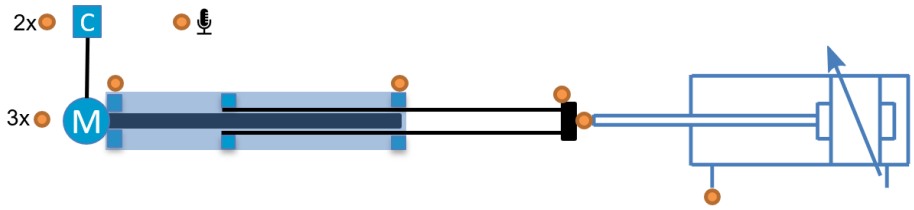


Figure 2.17: Schematic representation of the EMC, the ZeMA DAQ sensors localization (orange), and the pneumatic cylinder that simulates a load on the EMC (adapted from Paper 1, [146]).

One working cycle lasts 2.8 s and consists of a forward stroke, a waiting time (150 ms), and a return stroke, as shown in Figure 2.18.

For ML, only one second of the return stroke phase is used as the velocity and load are constant in this period. Typically, a lifetime test consists of more than 500,000 working cycles or more than 16 days. This lifetime is significantly shorter than the lifetime of an EMC in industrial applications, as the EMC is tested in the test bed under conditions that cause extreme wear. To make the data set more manageable, only every 100^{th} working cycle is included for each lifetime test, and the data is downsampled to 2 kHz. The cycle reduction only affects the number of cycles per target class. Thus, the individual lifetime tests for the three EMCs (axis 3, axis 5, and axis 7) consist of 6,292, 6,083, and 5,732 cycles. The downsampling has no significant influence on the ML results, as shown in Section 3.4. The measurement values in the EMC data set are stored

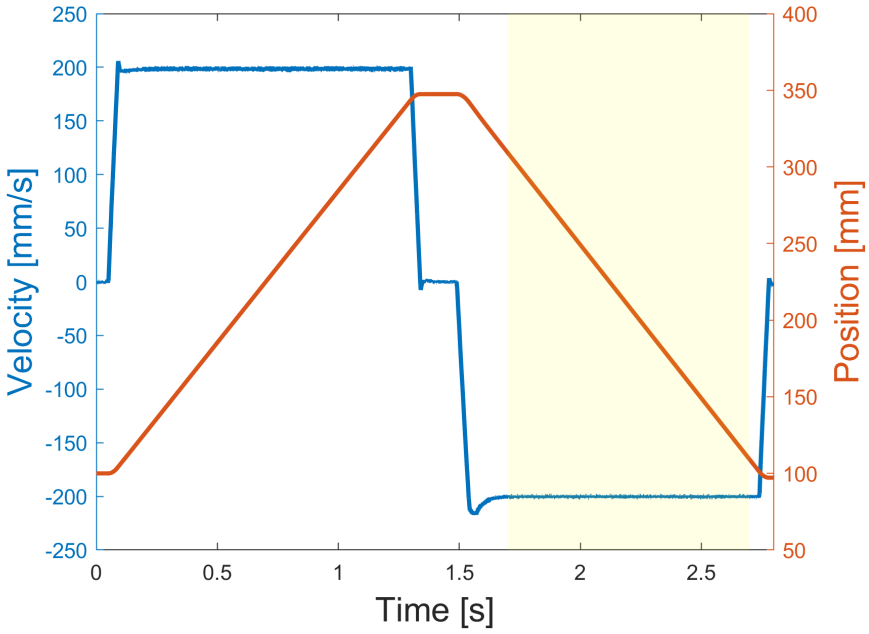


Figure 2.18: Velocity (blue) and position (red) of the EMC during one cycle. The yellow box marks one second of the return stroke that is used for ML (adapted from Paper 1, [146]).

as analog-to-digital converter (ADC) values. In addition to the data file containing the measurement values, a further document, including the conversion formula and its specific constants (e.g., gain and offset) per sensor, is provided on Zenodo [142] to convert the ADC values to *Système international d'unités* (SI) units.

2.3.3 Gas sensor data set

Humans' most crucial environment is the indoor environment, as they spend most of their lifetime indoors [147–149]. Therefore, their health, well-being, and performance are related to IAQ [150, 151] and can be negatively affected by polluted indoor air. As major pollutants in poor indoor air, volatile organic compounds (VOCs) can lead to serious health problems, e.g., sick building syndrome [152, 153], or even severe diseases, e.g., different cancer types [154, 155]. On the one hand, common sources of VOCs are humans themselves by exhalation and dermal emission [156, 157]. Therefore, the concentration of carbon dioxide (CO_2) emitted by humans inside a building can be used to approximate the IAQ there [158], and IAQ monitoring is mainly based on CO_2 measurements today. However, on the other hand, human activities such as cooking, heating (especially burning stoves), household cleaning, and tobacco smoking, as well

as their consumer products, for example, furniture and carpets, emit VOCs [159]. Due to the construction materials used, even the building itself contributes to the VOC concentration [160].

To quantify VOCs in indoor air, metal oxide semiconductor (MOS) gas sensors are widely used due to their low cost, robustness, and high sensitivity [161, 162]. However, MOS gas sensors have only a low selectivity. To improve their limited selectivity, e.g., to a group of gases like VOCs, they can be operated in dynamic modes, especially temperature modulation. This temperature modulation is also known as temperature cycled operation (TCO), which has first been proposed in 1974 [163] and further investigated and improved during the last decades [162, 164–167]. Using MOS gas sensors in TCO, i.e., a cyclical change of the sensing layer temperature, results in rich and extensive sensor response patterns, which can be interpreted using ML [162].

The gas sensor data set [168] used in this thesis was published in 2021 [167, 169]. It consists of several calibration measurements of random gas mixtures in a gas mixing apparatus (GMA) and several field test measurements of ambient air carried out in an office with the multilayer MOS gas sensor SGP30 [170], which has four sensitive layers on one hotplate [171]. As single VOCs for the calibration measurements in a GMA, four widely-used gases are chosen: acetone, ethanol, formaldehyde, and toluene. Formaldehyde is identified as one of the most toxic and carcinogenic gases in indoor air [172–174] and was listed under the five most hazardous gases by the INDEX project [175]. The most significant sources are pressed wood products, e.g., particleboard and plywood paneling [176]. Even toluene (e.g., in inks and paints [177]) is listed under the 13 most hazardous gases within the INDEX project [175]. The overall VOC concentration $\mathrm{VOC_{sum}}$ is calculated as the sum of the four VOC concentrations. Together with water vapor and inorganic background gases, i.e., hydrogen and carbon monoxide, they are the basis of the gas composition within the data set, as shown in Figure 2.19.

In the calibration period, the concentrations of seven different gases that are relevant for indoor air quality and humidity are randomly chosen from the ranges shown in Table 2.3 using *Latin hypercube sampling* (LHS) [179, 180] to obtain unique gas mixtures (UGMs).

The used temperature cycle (TC) consists of ten temperature steps at 400 °C with a duration of 5 s each. Every 400 °C phase is followed by a single constant low-temperature step of a level between 100 °C and 375 °C with a duration of 7 s each. Thus, one single temperature cycle lasts 120 s and consists of 2,400 measurement values for each gas-sensitive layer of the SGP30 (sampling rate 20 Hz). One TC represents one observation

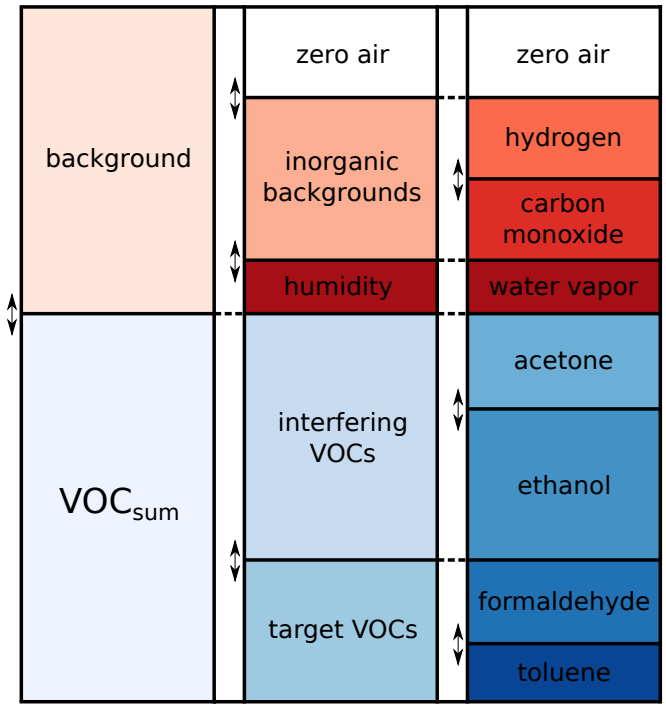


Figure 2.19: Overview of the gas composition consisting of background gases (red) and volatile organic compounds (blue) (adapted from [58, 178]).

(cycle) for each gas-sensitive layer, i.e., one row in each of the four data matrices $\mathbf{D}_i$, $i = 1, \ldots, 4$. The sensor response of the first gas-sensitive layer of the used SGP30, i.e., the logarithmic resistance, is shown for one cycle in Figure 2.20.

In this thesis, only the initial calibration period of the gas sensor data set is used. During this period, the SGP30 sensor is exposed to each UGM for ten TCs. The initial calibration period consists of 500 UGMs, resulting in 5,000 cycles. The limited time response of the GMA and synchronization problems between GMA and the MOS gas sensor SGP30 lead to an omission of the first four TCs and the last TC for each UGM, i.e., only five TCs are left per UGM. Due to run-in effects, the three UGMs at the beginning of the initial calibration period are also not considered. Thus, the used part of the gas sensor data set comprises 2,485 TCs during 497 UGMs with stable gas concentrations from the initial calibration. The gas sensor data set is suitable for the regression version of the AMLT, as gas concentrations are continuous natural values.

Table 2.3: Concentration ranges for all gases during the initial calibration period [181].

Substance	**Minimum**	**Maximum**
Hydrogen	400 ppb	2,000 ppb
Carbon monoxide	150 ppb	2,000 ppb
Humidity	25 % RH	70 % RH
Acetone	14 ppb	300 ppb
Ethanol	4 ppb	300 ppb
Formaldehyde	1 ppb	400 ppb
Toluene	4 ppb	300 ppb
VOC_{sum}	300 ppb	1,200 ppb

RH: relative humidity

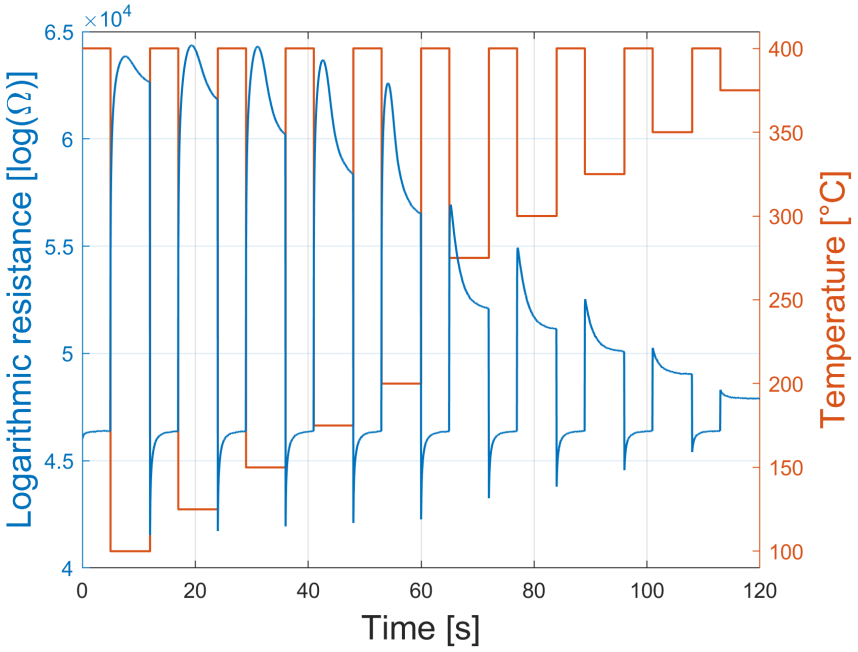


Figure 2.20: Response of the first gas-sensitive layer for one UGM of the SGP30 (blue) during the used TC (red).

2.4 FAIR data

The data sets presented in Section 2.3 have one major disadvantage. In addition to the file containing the data itself, one or more additional files are required, which contain further information, e.g., explanations of the variable names, references to publications concerning the data set, or even conversion formulas from ADC values to SI units. This additional information often poses problems for users of the data set if they are not

included in the data set. To obtain all the relevant information concerning the data set, necessary explanations must be provided in at least one additional file. Therefore, good data management is required to reuse already existing data sets to provide long-term data storage, thus consequently achieving sustainability [182–184]. Data sharing and exchanging in research and industry can benefit scientific progress. Moreover, given the increasing advancement of digitalization, the data usage process should be automated, which leads to the fact that machines' ability to find and use a data set must be improved, resulting in the need for machine-readable metadata within a data set.

In 2016, a consortium of scientists and organizations published the FAIR data principles and their 15 subprinciples [185]. These FAIR principles provide guidelines to enhance **F**indability, **A**ccessibility, **I**nteroperability, and **R**eusability of digital resources, such as code and data sets. Their primary focus is machine readability, i.e., the ability of machines, e.g., computers, to find, access, interoperate, and reuse digital resources without human intervention or only with minimal human assistance. In 2021, 66 % of researchers surveyed had heard of the principles, however, only 28 % of them were familiar with the principles [186], although the FAIRness of data is highly relevant. Making data traceable and FAIR is a manageable burden for the individual researcher. It reduces administrative effort and saves time when reusing the data, e.g., by another institute, a colleague, or the original researcher himself.

Reaching FAIRness involves three areas: the data themselves, the metadata describing these data, and the necessary infrastructure, e.g., the data storage. Yearly, thousands of petabytes of data are collected, and their potential cannot be realized due to missing FAIR compliance [187].

First and foremost, (re)using data means finding them. Therefore, digital objects as a composition of data and machine-readable metadata with unique and persistent identifiers, e.g., a *Digital Object Identifier* (DOI), are essential for machines and humans to find the data. In the context of Open Science, the online access repository Zenodo [134] is one suitable archive with searchable metadata. Metadata should follow a common structure and terminology. Thus, semantic descriptors are used in the metadata to reach not only machine readability but also machine interpretability. Commonly used ontologies and knowledge representations are *Dublin Core* (DC) [188], *Digital System of Units* (DSI) [189], *Quantities, Units, Dimensions, and Types* (QUDT) [34], *Resource Description Framework* (RDF) [190], and *Semantic Sensor Network* (SSN) [35] which includes the *Sensor, Observation, Sample, and Actuator* (SOSA) ontology [191].

After finding the required data, knowledge about how the data can be accessed is necessary. Accessibility is often confused with Open Data, but in the context of FAIR data, this only means that there are clearly and transparently defined conditions for accessibility, e.g., access only for an individual research institute or access after authorization.

To realize data interoperability, the data must be enriched with machine-readable metrological properties such as types of physical quantities and the corresponding units of measurement. As all measurements are subject to uncertainty, measurement uncertainty provided by calibration, if available, should also be provided within a data set.

A good description of the data in terms of metadata is necessary for reusing data. These top-level metadata contain information about the data set, e.g., the creators of the data, the project in which the data was recorded, or the license of the data (e.g., *Creative Commons Attribution 4.0 International* (CC-BY-4.0)). To collaborate between different research institutes and industry partners, it is necessary to use common, open, and well-described data formats, e.g., *Hierarchical Data Format Version 5* (HDF5) for data and *JavaScript Object Notation* (JSON) for metadata.

For assessing FAIRness of a data set by its originator, i.e., a self-control if a data set achieves a certain level of FAIRness, the FAIR data maturity model is used [192, 193]. Forty-one measurable aspects concerning the data and the metadata, called FAIR data maturity model indicators, are defined for the four main FAIR data principles, as shown in [192]. Twenty indicators are classified as essential, 14 as important, and seven as useful. Each indicator can be evaluated by five levels [193]:

- not applicable (0),
- not being considered yet (1),
- under consideration or in planning phase (2),
- in implementation phase (3), and
- fully implemented (4).

These levels also give ideas on how the FAIRness level of the data can be improved and where to concentrate the effort. For visualization of the FAIRness level, four radar charts, one for each main FAIR data principle, are usually used.

3 Results and Discussions

This chapter contains detailed presentations and discussions of the published peer-reviewed papers that comprise the central part of this cumulative dissertation. An additional self-generated and annotated data set is presented, which fits the FAIR data principles.

3.1 Introduction

Industry 4.0 (I4.0), the fourth industrial revolution, denotes the transformation from traditional to smart manufacturing using digitalization. Smart factories, the so-called Factories of the Future (FoFs), allow increased flexibility in manufacturing and production, as well as better quality and improved productivity [194]. In FoFs, which represent the core elements of I4.0, the *Industrial Internet of Things* (IIoT) builds the networking basis for interconnected (smart) sensors, i.e., enabling the communication between individual sensors. In IIoT environments, often large numbers of sensors from a wide range of applications, e.g., micro-electro-mechanical systems (MEMS) sensors, are used due to their flexibility and cost-efficiency [31]. In a distributed sensor network, these sensors independently collect data from a wide range of different physical quantities from all levels of the manufacturing and production processes. This means that the sensor data does not necessarily have the same time basis. Therefore, sensor data fusion is required to bring the data of several sensors together in a consistent way.

In a survey presented in [195], data scientists spend 80 % of their time and effort on data preprocessing and data gathering. In contrast, only 20% of their time and effort are used for the actual machine learning (ML) analysis. Thus, data preprocessing, including labeling data, cleaning data, for example, from corrupt, duplicate, and incomplete data, fusion data, and organizing data, i.e., bringing data in a useful format for easily accessing and analyzing it, is an essential step in ML analysis projects. Data preprocessing is often not only time-consuming but also computationally complex leading to the idea of

providing a data pipeline for calibrated MEMS sensors to enhance and automate this process and, thus, reduce the required effort for the data preprocessing step.

Sensor data quality is essential to fully use the wide-ranging potential of smart sensors in the context of I4.0 and IIoT [196]. For example, time synchronization problems within a sensor network, sensor precision, sensor drift, or sensor failure can directly influence the sensor data quality. Considering uncertainty in both time and value is a key element for obtaining reliable data and performing ML with reliable data. Therefore, it is required to have appropriate data sets, preferably data sets that meet the FAIR principles and already contain associated measurement uncertainties to the measured values. FAIRness and metrological traceable data build a basis for data exchange in research and industry.

However, measurement uncertainty evaluation for each prediction of an ML model is often neglected in the ML process [26]. Thus, measurement uncertainty for raw sensor data is not or only rarely included in published data sets. On the one hand, uncertainty information can be obtained from manufacturers' datasheets; however, these datasheets only give rough estimates for the uncertainty values. On the other hand, the more suitable but expensive way is to derive uncertainty values based on dynamic calibration information. Traceability in the FoF is achieved by consideration of measurement uncertainty from the calibration of individual sensors through to data-driven ML analysis [31].

Confidence in ML algorithms, their decisions, and their predictions is crucial. Quantitative measurements are carried out in industry every day, and as no measurement result is exact, measurement uncertainty occurs in both time and value. In distributed sensor networks, one of the causes for uncertainty in time can be time synchronization errors between individual sensors, for example resulting from network communication problems or time delays. Performing sensor data fusion in the correct way is crucial for ML applications as this directly influences the ML model performance. Uncertainty in the measurement value results from sensor degradation over time, limitation of the measurement systems, or simply the use of non-calibrated or low-performance sensors. Considering uncertainty information in time and value for performing ML contributes to the data quality, and therefore influence on the model-based ML results can be expected. Thus, determining measurement uncertainty and using it in ML must not be regarded as an additional burden; instead, it is a worthwhile addition with added value.

3.2 Paper 1 – Uncertainty-aware data pipeline of calibrated MEMS sensors used for machine learning

Industry 4.0 requires a digitalized factory, the so-called FoF, where (smart) sensors are used to increase competitiveness and efficiency [8]. In Factories of the Future, using a large number of low-cost sensors is preferred over a small number of high-quality and expensive sensors to reduce cost or increase robustness through redundancy [31]. Intensive data preprocessing with a lot of challenges is required to obtain high-quality preprocessed data from low-cost sensors as their origin data quality is limited. Improving the data quality is necessary, as the quality and accuracy of ML results is directly related to the data quality [197, 198].

In Paper 1 [146], the starting point is an already existing test bed for condition monitoring and lifetime tests of electromechanical cylinders (EMCs) at *Zentrum für Mechatronik und Automatisierungstechnik gGmbH* (ZeMA). As described in Section 2.3.2, the test bed is equipped with eleven sensors with sampling rates between 10 kHz and 1 MHz. Data acquired with this expensive ZeMA data acquisition unit (DAQ) system has already been successfully used for wear detection and lifetime prediction of electromechanical cylinders by applying ML [37, 139].

To obtain data from low-cost sensors, the test bed for EMC lifetime tests (c.f. Section 2.3.2) is equipped with a further DAQ module based on the microcontroller STM32F767ZI [199], the so-called *Smart-Up Unit* (SUU) [36, 200], which has been developed in the project *Metrology for the Factory of the Future* (Met4FoF) [31]. The SUU uses three MEMS sensors:

- a 3-axis accelerometer (Bosch BMA 280 [201]) with a sampling rate of 2 kHz,
- a 9-axis inertial measurement unit (InvenSense MPU 9250 [202]) with a sampling rate of 100 Hz for the magnetic flux density and 1 kHz for the acceleration as well as for the angular speed, and
- a combined pressure and temperature sensor (TE Connectivity MS 5837-02BA [203]) with a sampling rate of 1 Hz.

These three sensors are located at the plain bearing on the same sensor holder as the plain bearing accelerometer of the ZeMA DAQ. Additionally, independent absolute timestamps based on the *Global Navigation Satellite System* (GNSS), especially *Global Positioning System* (GPS) in the presented paper, are stored for every measurement

value. These absolute timestamps have nanosecond resolution and sub-microsecond uncertainty, as the used GNSS module provides a time reference synchronized with the SUU hardware timer via a pulse-per-second (PPS) signal. To investigate the potential of the low-cost SUU, a lifetime test of an EMC was carried out and time series data with both systems, the ZeMA DAQ and the SUU, were acquired.

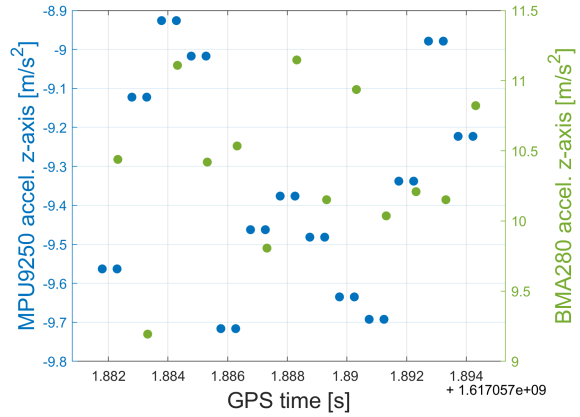
In contrast to the ZeMA DAQ, which only records working cycles of 2.8 s, the SUU acquires data continuously. To use ML methods, ZeMA DAQ and SUU data need to be brought together consistently so that the SUU data fits the shape of the ZeMA DAQ data. For this sensor data fusion process, a data processing pipeline for the SUU data is proposed, which begins with the raw SUU sensors data and the calibration data of these sensors, and after several steps, leads to a cycle-wise data set with corresponding uncertainty values. The trigger signal of the ZeMA DAQ, which indicates the start of each working cycle, is the only link between the ZeMA DAQ and the SUU. It is not only recorded by the ZeMA DAQ but also by the SUU and builds the basis for the data alignment process. Cycles in the SUU data are detected and extracted with the rising edge of the trigger signal.

During the data preprocessing step, several unforeseen issues occurred, which have to be threatened. Accidentally, the acceleration sampling rates of two sensors (InvenSense MPU 9250 and Bosch BMA 280) were interchanged in the software version of the SUU used for data acquisition. As the MPU 9250 can only be sampled with 1 kHz, a 2 kHz sampling rate leads to two identical measurement values at different time stamps, as shown in Figure 3.1a. To solve this problem, only every second measurement value of the MPU 9250 accelerations with its corresponding GPS time stamp is used in Paper 1. In the current software version [205], this bug is corrected.

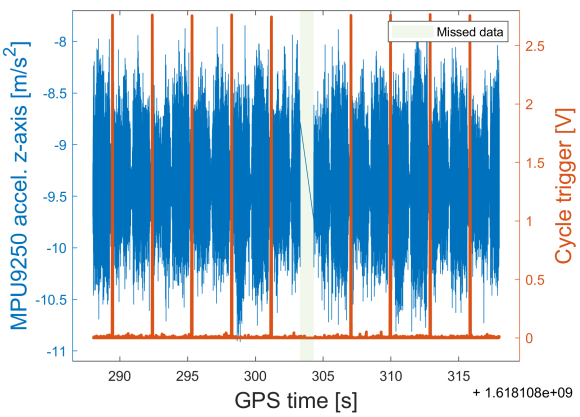
As the data is stored in multiple files every 30 min, a file switch leads to a data loss of approximately 1 s, as shown in Figure 3.1b. If this problem occurs, the affected cycle is ignored, and the next one is taken as there is no noticeable difference in the signals of two directly following cycles.

Furthermore, due to hardware connection issues, some rising edges of the trigger signal are not recorded, although data from the sensors show that there should have been rising edges (cf. Figure 3.1c). Data, which is affected by the missing trigger, is made usable by halving the distance between the two rising edges around the missing one and assuming that the missing rising edge is located in the middle.

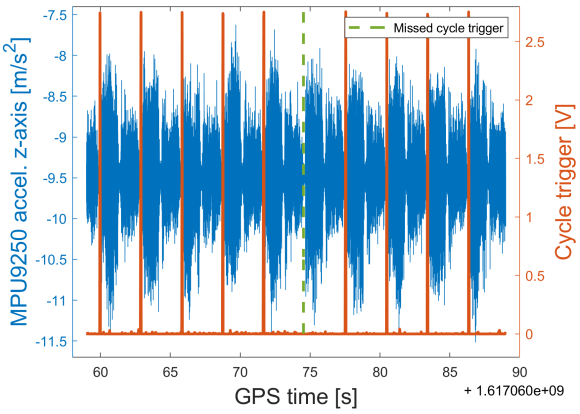
In the end, it was possible to detect 476,617 cycles in the SUU data compared to 476,560 in the ZeMA DAQ data. The higher number of cycles in the SUU data can be



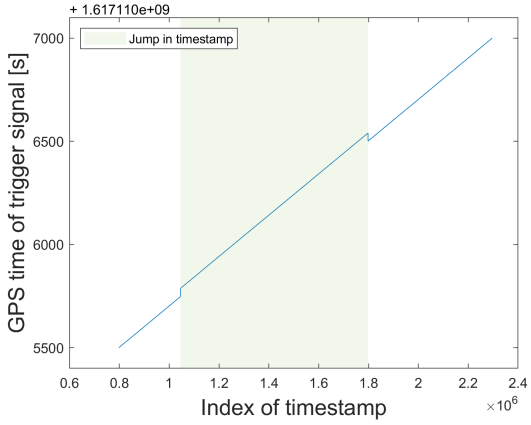
(a) Interchanged sampling rates of the MPU 9250 (blue) and BMA 280 (green), meaning only every second measurement value is different for the MPU 9250 [204].



(b) MPU 9250 acceleration data in one direction (blue), the trigger signal (red), and the missing data between two recorded files (green area) [204].



(c) MPU 9250 acceleration data in one direction (blue), the trigger signal (red), and the missing rising edge (dashed green) [204].



(d) Time glitch of approx. 40 s. The green area is fitted using a robust linear regression [204].

Figure 3.1: Issues in the data preprocessing step.

explained by the fact that the ZeMA DAQ only records data when it receives a trigger signal.

Further issues are time glitches, leading to forward and backward jumps of integer multiples of 1 s in 11.38 ppm of the timestamps, which are not handled in the used SUU software version. Such a time glitch is shown in Figure 3.1d. These jumps are detected by a robust linear regression of timestamps over the sample index. Assuming that less than 50 % of the timestamps in one file are affected by a time glitch [206], timestamps are replaced by regressed values if their difference is larger than a chosen threshold. These time glitches are treated in the current software version [205] and, thus, will be

no issue anymore in the future. Correct timestamps are required for the next step of SUU data preprocessing.

To make the extracted working cycles equidistant, interpolation is necessary, and uncertainty must be propagated through the interpolation algorithms. Five interpolation methods are considered: linear and cubic spline, as well as next, nearest, and previous neighbor. For the different interpolation methods and their uncertainty propagation, the Python package PyDynamic is used [207, 208]. In this software package, the calculation of the sensitivity coefficients for the different interpolation methods is based on the *Guide to the Expression of Uncertainty in Measurement* (GUM) and carried out according to the corresponding uncertainty propagation formulas provided by White et al. [209, 210].

As uncertainty values based on manufacturer's datasheets are only rough estimates, uncertainty values derived from dynamic calibration information are used in Paper 1. Paper 1 only shows how to obtain measurement uncertainty values from dynamic calibration information, whereas consideration of measurement uncertainty and the influence on ML results is investigated in Paper 4. Dynamic calibration is performed using sine excitation at different frequencies and amplitudes under consideration of absolute timestamps [36]. As references, the velocity of the mechanical excitations is measured with three laser Doppler vibrometers. To represent the inverse transfer behavior, a stable infinite impulse response (IIR) filter is chosen. The numerator and denominator coefficients of a stable IIR filter are determined, and the uncertainties associated with the filter coefficients are calculated using the PyDynamic function `invLSIIR_unc` [211]. This function propagates uncertainty based on a Monte Carlo method described in Supplement 2 to the GUM (GUM-S2) [44]. To obtain the uncertainty values associated with the measurement values, the IIR filter with uncertainty is applied to the raw sensor readings [212].

For performing ML with the automated machine learning toolbox (AMLT), ZeMA DAQ and SUU data are used. Using 1 % lifetime increments as target and comparing data downsampled to 1 kHz for both systems, the ZeMA DAQ data set leads to more precise and accurate lifetime predictions than the SUU data set. There are also performance differences between the different interpolation schemes used for the data of the SUU. Next and previous neighbor perform 10 % worse than linear and cubic spline, as well as nearest neighbor, which performs nearly equally.

However, using a lower resolution for the remaining useful lifetime (RUL) estimation, i.e., 10 % increments which means approximately 39 h instead of 1 % (3.9 h), the cubic interpolated data of the SUU leads to a classification error of 45.95 %. In comparison,

the ZeMA DAQ data achieves a classification error of 26.47 % using 1 % increments for the lifetime target. As the classification error only considers whether the correct target class is predicted or not, the root-mean-square error (RMSE) is used, which takes the distances between the actual and the predicted target values into account. The RMSE for the ZeMA DAQ data is 1.39 %, whereas it is 5.25 % for the SUU data. This shows that, in the light of "fitness for purpose," the SUU, which is of much lower cost than the complex ZeMA DAQ, can also provide suitable results for the RUL estimation of the EMC. However, the estimated lifetime does not have the same resolution as for the ZeMA DAQ data, but it will likely be sufficient for most maintenance interval scheduling tasks.

Measurement: Sensors 22 (2022) 100376

Contents lists available at ScienceDirect

Measurement: Sensors

journal homepage: www.sciencedirect.com/journal/measurement-sensors

Uncertainty-aware data pipeline of calibrated MEMS sensors used for machine learning

Tanja Dorst [a,b,*], Maximilian Gruber [c], Benedikt Seeger [c], Anupam Prasad Vedurmudi [c], Tizian Schneider [a,b], Sascha Eichstädt [c], Andreas Schütze [a,b]

[a] *ZeMA – Center for Mechatronics and Automation Technology gGmbH, Saarbrücken, Germany*
[b] *Lab for Measurement Technology, Department of Mechatronics, Saarland University, Saarbrücken, Germany*
[c] *Physikalisch-Technische Bundesanstalt, Braunschweig, Berlin, Germany*

ARTICLE INFO

Keywords:
Machine learning
Dynamic measurement uncertainty
Interpolation
Time series
Predictive maintenance
Low cost sensor network

ABSTRACT

Sensors are a key element of recent Industry 4.0 developments and currently further sophisticated functionality is embedded into them, leading to smart sensors. In a typical "Factory of the Future" (FoF) scenario, several smart sensors and different data acquisition units (DAQs) will be used to monitor the same process, e.g. the wear of a critical component, in this paper an electromechanical cylinder (EMC). If the use of machine learning (ML) applications is of interest, data of all sensors and DAQs need to be brought together in a consistent way. To enable quality information of the obtained ML results, decisions should also take the measurement uncertainty into account. This contribution shows an ML pipeline for time series data of calibrated Micro-Electro-Mechanical Systems (MEMS) sensors. Data from a lifetime test of an EMC from multiple DAQs is integrated by alignment, (different schemes of) interpolation and careful handling of data defects to feed an automated ML toolbox. In addition, uncertainty of the raw data is obtained from calibration information and is evaluated in all steps of the data processing pipeline. The results for the lifetime prognosis of the EMC are evaluated in the light of "fitness for purpose".

1. Introduction

Industrial processes are typically monitored by processing time series data acquired by (smart) sensors. In the field of Industry 4.0, *machine learning* (ML) methods have become a popular choice to extract features of interest from raw time series signals. Although not a limitation of ML in general, many of these algorithms dealing with time series data expect input data with equidistant timestamps. Data acquisition is often performed by microcontrollers at a sampling frequency derived from an internal oscillator. As these oscillators experience general variances arising from factors such as temperature dependence, the yielded sample times differ from the desired values.

The traditional approach is to sample all necessary sensors using the same (multi-channel) *data acquisition unit* (DAQ). In this case, the common assumption is that all data is equidistant in time (within some margin of error). However, in Industry 4.0 scenarios data is acquired using multiple independent DAQs and the assumption of equidistant timestamps is prone to failure. As a consequence of the accumulated sample-frequency drift, the i-th record of one smart sensor is no longer guaranteed to represent the same moment in time as the i-th record of another one. Recent studies [1] showed that this can lead to significant errors in a subsequent ML processing pipeline.

Moreover, to obtain reliable data, it is of interest to consider uncertainty as defined in Refs. [2,3] both in time and value. In the considered setting, timestamps with uncertainty are obtained from a smart sensor system. In general, uncertainty associated with the measurand can be derived from the manufacturer's datasheet or calibration information. In this contribution, it is shown how uncertainty values can be derived based on dynamic calibration information, as manufacturer's datasheets only give rough estimates of the uncertainty values.

To provide the same equidistant time base it is necessary to synchronize data from two or more independent sources. After performing this alignment, the influence of multiple interpolation schemes on subsequent ML pipelines and on the associated processing steps is investigated. The proposed solution is applied to data obtained by a test bed for life-time prognosis and end-of-line tests of *electromechanical cylinders*

* Corresponding author. Lab for Measurement Technology, Department of Mechatronics, Saarland University, Saarbrücken, Germany.
E-mail address: t.dorst@lmt.uni-saarland.de (T. Dorst).

https://doi.org/10.1016/j.measen.2022.100376
Received 27 January 2022; Received in revised form 19 April 2022; Accepted 2 May 2022
Available online 6 May 2022

(EMCs) with an existing analog sampling DAQ. In a recent measurement campaign, the test bed was observed with the original DAQ system and an additional smart sensor (combining multiple digital sensors) with independent absolute timestamping based on the *Global Navigation Satellite System* (GNSS). An overview of the required pipeline is shown in Fig. 1. While providing references to the specific sections the figure also illustrates that most of the effort is included in the data preprocessing steps.

2. Measurement setup

The used data set is generated by a test bed for lifetime tests of EMCs. The main components of the test bed are the EMC under test (Festo ESBF cylinder [4]), schematic shown in Fig. 2, and a pneumatic cylinder simulating a load of 7 kN, equivalent to the maximum load according to manufacturer specifications, on the EMC in axial direction. Fig. 3 shows the scheme of the test bed.

A typical working cycle, which can be seen in Fig. 4, consists of a forward stroke, a waiting time (150 ms) and a return stroke, and lasts 2.8 s.

Both linear movements of the EMC are always carried out at maximum speed and acceleration of approx. 200 mm/s and 5 m/s^2, respectively. In this test bed, long-term high load and speed driving tests are carried out until the EMC fails. Failure is determined by the end position accuracy criterion (< 30 mm deviation) which after degradation is no longer met due to increased friction. The typical lifetime of an EMC under these test conditions in earlier experiments was approx. 630,000 cycles or 20 days.

2.1. ZeMA DAQ characteristics of the EMC test bed measurement system

This test bed is equipped with eleven different sensors recorded with the test bed DAQ system:

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

The cycle-by-cycle data acquisition of the EMC test bed is triggered by a digital output of the motor controller which is parameterized via the proprietary *Festo Configuration Tool* (FCT) software to provide an edge signal when the motion profile starts. The data acquisition at the EMC test bed is carried out with a NI PXI system with three modules (cf. Fig. 5):

- Reconfigurable oscilloscope PXIe-5170R with eight simultaneously-sampled channels, up to 250 MS/s and 14 bit resolution [7],

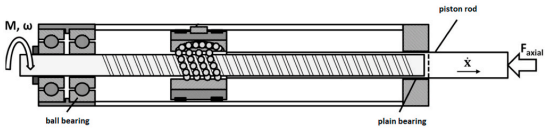


Fig. 2. Simplified structure of an EMC with a spindle drive (adapted from Ref. [5]).

- Sound and vibration module PXIe-4492 with eight simultaneously-sampled channels, up to 204.8 kS/s and 24 bit resolution [8], and,
- Multifunction I/O module PXIe-6341 with eight differential or 16 single-ended channels, up to 500 kS/s and 16 bit resolution [9].

2.2. SmartUp unit characteristics

The *SmartUp Unit* (SUU) is a DAQ module based on an STM32F767Zi microcontroller [10]. It is installed in parallel to the original DAQ and was first deployed as part of a method to demonstrate the calibration of a digital Micro-Electro-Mechanical Systems (MEMS) sensor [11]. The SUU is capable of connecting to the digital interfaces of modern integrated sensors via SPI or I^{2}C and transmit the data generated as a time series with hardware generated timestamps. The integer values of the digital sensors are converted by the SUU with the nominal scaling factors into SI floating point values. The SUU also provides meta information on the measured values such as full-scale range, resolution (e.g. 2^{16} *least significant bits* (LSBs)), measured quantity (e.g. "`X Acceleration`") and units (e.g. "`\metre\second\tothe{-2}`" in accordance with [12]) in a stateless protocol. As part of the *Metrology for the Factory of the Future* (Met4FoF) project [13], the SUU is also designed for use in Industry 4.0 environments. A key application in Met4FoF is condition monitoring which necessitates the availability of reliable time-synchronized data [14]. The SUU enables time-synchronization by means of a GNSS receiver. The GNSS module provides a time-reference which is synchronized with the hardware timers of the SUU via a *pulse-per-second* (PPS) signal resulting in an absolute timestamp with nanosecond resolution and sub-microsecond uncertainty. The PPS signal is generated by atomic clocks that are part of the GNSS, leading to this high accuracy.

The SUU is equipped with three digital sensors

- a 9-axis inertial measurement unit (InvenSense MPU-9250) [15],
- a 3-axis accelerometer (Bosch BMA280) [16], and
- a combined pressure and temperature sensor (TE Connectivity MS 5837-02BA) [17],

such that three sensors and the SUU together form a smart sensor. As shown in Fig. 6, the sensors of the SUU are placed on the same sensor holder as the acceleration sensor Kistler 8712A5M1 [18] at the plain bearing of the ZeMA test bed.

Fig. 7 summarizes the sensors of the SUU (purple dots) and the sensors of the ZeMA DAQ unit (red dots) as well as their location with respect to the EMC. The green triangle symbolizes the trigger signal of

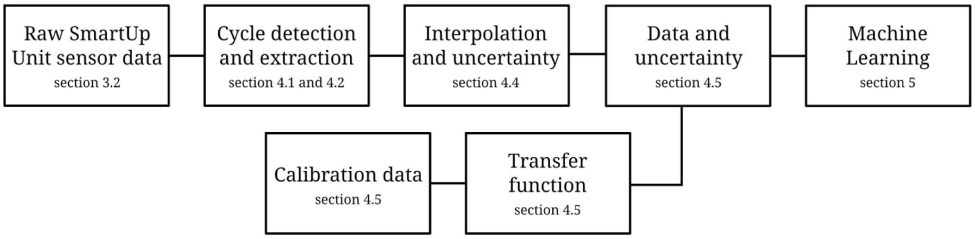


Fig. 1. Overview of the data processing pipeline.

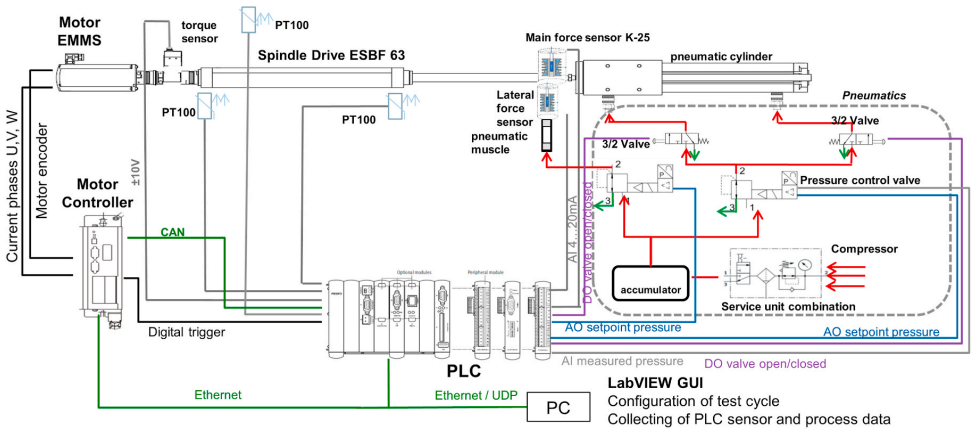


Fig. 3. Scheme of the test bed [6].

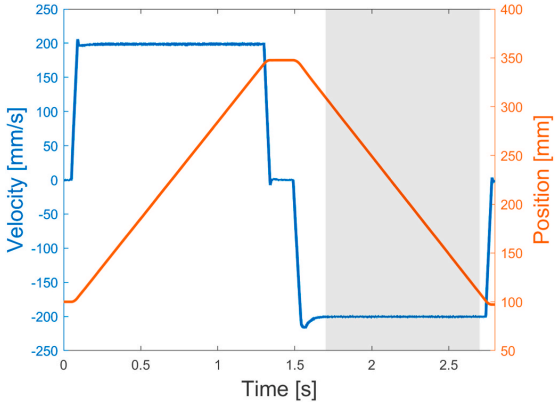


Fig. 4. Working cycle of the EMC test bed.

the ZeMA test bed which is recorded by the SUU and is hence the only link between both DAQ units. In the following, the MS 5837-02BA sensor is ignored as this sensor had a defect after its installation on the test bed.

3. Data presentation

The data recorded on both DAQs represent the same EMC lifetime test executed in April 2021 which lasted approx. 16.5 days (387.83 h).

3.1. ZeMA DAQ

The raw data set generated by the ZeMA DAQ totals to 9.41 TB from 476,560 cycles. The recording is event-centric and records 2.8 s of every cycle. As a consequence, a small period (~ 0.1 s) between cycles is omitted. The acquisition does not record any absolute time information, but is assumed to be equidistant in time.

To save computational costs in later processing steps, a data set downsampled to 2 kHz is created. This includes only 1 s of the return stroke (gray area in Fig. 4) of every 100th cycle beginning at cycle 51, bringing the size down to less than 2 GB. The data recorded during cycle 51 with the eleven sensors of the ZeMA DAQ is shown in Fig. 17).

3.2. SUU DAQ

During the same test the SUU generates a data set with 71 GB. The recorded data is not event-centric, but consists of continuous recordings split across multiple files. The acquisition uses GNSS to provide absolute timestamp information for the acquisition time of every datapoint. Due to the temperature dependence of the internal oscillator's behavior of the sensor [11], the time series is non-equidistant. The data recorded during cycle 51 with the BMA 280 and the MPU 9250 SUU is shown in Fig. 18 and in Fig. 19, respectively. In addition to the main sensors of the SUU described in section 2.2, a trigger signal from the ZeMA DAQ is recorded. This trigger marks the start of a new cycle in the ZeMA system by providing a voltage signal with a short peak above 2.5 V.

3.3. Data preprocessing strategy

Methods for feature extraction and selection based on the event-centric data structure have been described recently [19,20]. In order

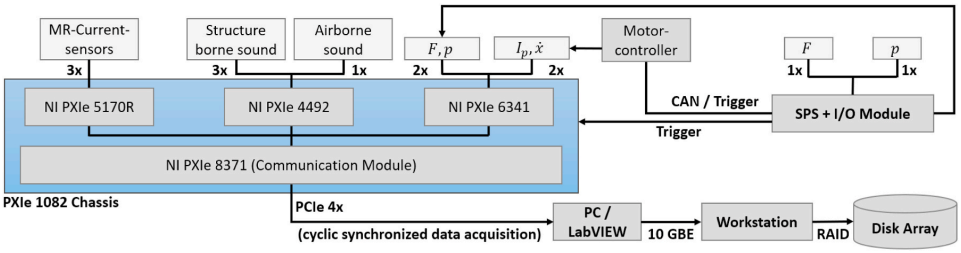


Fig. 5. Data acquisition unit of the EMC test bed (adapted from Ref. [5]).

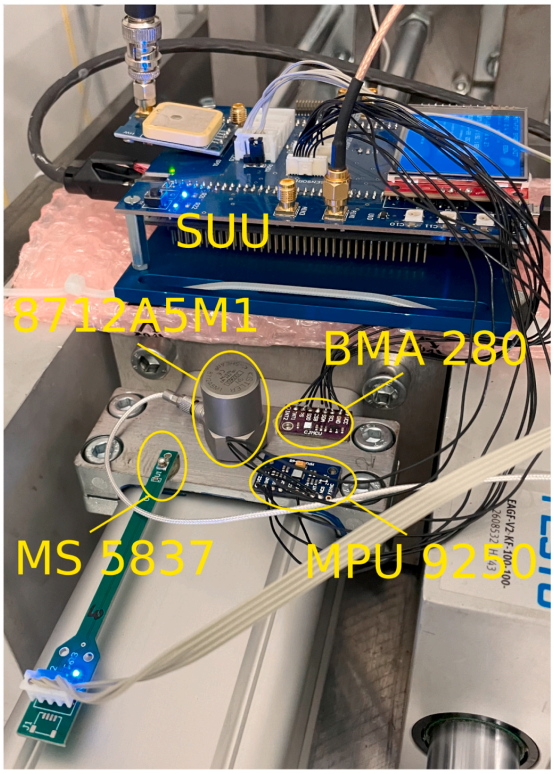


Fig. 6. Installation of the SUU and its three sensors in the EMC test bed. The Kistler 8712A5M1 acceleration sensor of the ZeMA DAQ is also installed at the same location.

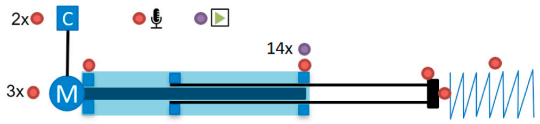


Fig. 7. SUU sensors (purple) and ZeMA DAQ sensors (red) localization with respect to the EMC. The green triangle symbolizes the trigger, which indicates the start of a cycle and is recorded by both the ZeMA DAQ and the SUU. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

to use the existing methods in conjunction with the smart sensor data from the SUU, it is necessary to extract event-centric data from the SUU data set and save it in a format similar to the downsampled ZeMA DAQ data set. To bring the data of two independent sources together, the following steps are necessary:

1. The data needs to be aligned temporally. This includes the challenge of establishing a conversion between the implicit relative timestamps of the original DAQ and the absolute timestamps of the smart sensor by an analysis of a common signal. This enables a correction of the drift of the original DAQ-time base, as well as a quantification of its time uncertainty.
2. Both data sets need to be represented on the same equidistant time base. This is achieved by interpolation of the SUU data set.

The temporal alignment of both raw data sets is achieved by cycle detection and an appropriate bookkeeping. It is then possible to extract time series of specific cycles and interpolate them to equidistant time matching the representation used in the downsampled ZeMA data set.

These steps are described in more detail in the following section. The preprocessed data set is available as a standalone publication with comprehensive annotations [21].

4. Methods for ZeMA DAQ and SUU data alignment

In order to represent the data recorded by the SUU in the same event-centric structure as used by the ZeMA DAQ, certain methods need to be applied. An overview of the pipeline steps is already given in Fig. 1. The proposed cycle detection, extraction and interpolation is shown for an exemplary time period in Fig. 8 and detailed in sections 4.1 and 4.2. In section 4.3, issues encountered with the recorded timestamps are fixed to allow successful interpolation. Uncertainty-aware data processing is an enabler of metrological traceability. Therefore, the uncertainty for the interpolation and the uncertainty for the calibration-based compensated raw data are described in sections 4.4 and 4.5, respectively. The effect of timestamp uncertainty was also investigated, but was found not to be relevant in the presented setup. However, a short justification is given in A.

4.1. Cycle detection

The general idea to detect cycles in the SUU data set is based on the recorded trigger signal. Rising edges in this signal mark the beginning of a new cycle. This allows detection of the start of a new cycle with an uncertainty of $u(t_{0,i}) = 1$ ms corresponding to the sample rate of the trigger signal.

Directly applying this method yields fewer than the expected number of total cycles but some of them have double the typical length. A detailed inspection of the raw data identified three main causes for the presumably “missing triggers”:

1. rising edge in the first entry of a file
2. rising edge starts and ends between two files
3. no rising edge, but data on other channels indicate a cycle

Problems (i) and (ii) are caused by data aggregation in multiple files, each typically storing the datapoints of 30 min. Because of the nature of the overall acquisition pipeline, switching to the next file takes 1.5 s during which no data is recorded. Problem (iii) is likely independent of the SUU and related to hardware connection issues but no further investigation was performed in this contribution.

The first problem can be handled by a consistent bookkeeping of the last datapoint of the trigger signal in the previous file. The second and third problems, although differing in cause, can both be handled similarly. If the duration between two cycle starts exceeds 4 s, it is assumed that a previously undetected cycle started in the center between both surrounding triggers. Furthermore, to detect triggers that might fall into the short “blackout” period between two files, the last trigger of the previous file is included.

By considering these special cases, 476,617 cycles are detected in the SUU data compared to 476,560 in the ZeMA data. The main cause for the difference is that the ZeMA DAQ only records data when the trigger signal is received. As mentioned in problem (iii), the SUU also detects cycles, when no trigger is encountered by inserting “virtual” triggers. This leads to a slightly higher number of detected cycles for the SUU.

4.2. Cycle extraction

Detected cycles are numbered in ascending order across files and every 100th cycle starting from the 51st cycle (1-based array indexing) gets extracted. The time series of a cycle of interest are all data points

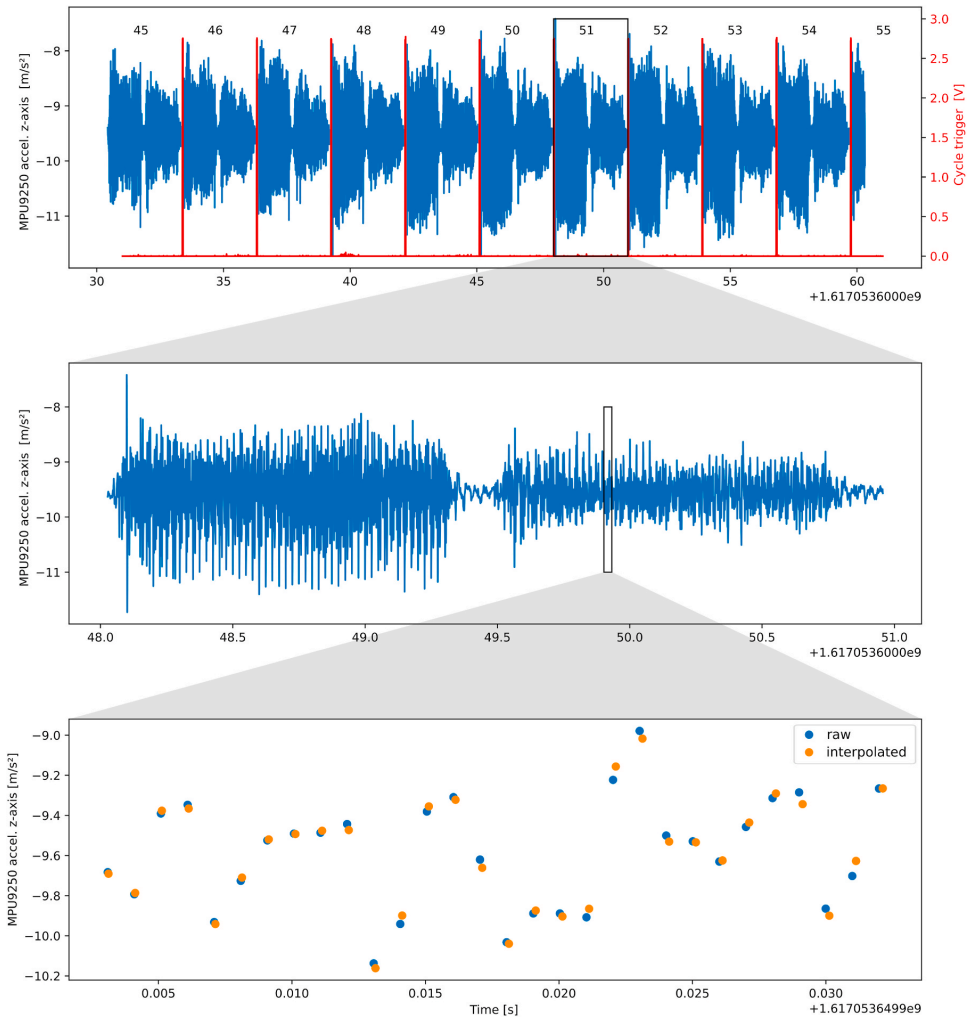


Fig. 8. Breakdown of preprocessing steps. Top: Extract cycle count from trigger signal. Mid: extracted 51 cycle. Bottom: detail of interpolation at given sample rate.

recorded between the start of the cycle (included) and the start of the next cycle (excluded). If the time series of the cycle of interest does not allow the interpolation of the return stroke (e.g. missing data because of file end), the next cycle is chosen instead. Uncertainty of data is quantified as half a LSB, which is an optimistic estimation.

4.3. Time Glitch treatment

In early versions of the SUU software, the *Global Positioning System Fix Data* (GGA) message was evaluated after the PPS pulse, without checking whether there was really a new fix. If only two satellites are in the field of view, there is a PPS pulse but no new fix. This led to leaps of integer multiples of 1 s in the measurement data. Although the issue is fixed in the latest version of the SUU software, some of the acquired data required correction in the preprocessing pipeline. The jumps forwards and backwards in time are detected by a robust regression of timestamps over the sample index. Timestamps differing more than a threshold from the fitted regression line are replaced with the regressed values under the assumption that less than 50% of the data in a file is corrupted by the glitch [22]. On average, time glitch treatment was required for only 11.38 ppm of the recorded data.

4.4. Uncertainty of interpolation

Extracted cycles are made equidistant by relying on the Python package `PyDynamic` which provides the method `interp1d_unc` and propagates uncertainty based on [2,23,24].[1] The uncertainty of a spline interpolation is calculated with

$$\widehat{y}(t) = \sum_{i=1}^{N} y_i F_i(t, t_1, \ldots, t_N) \tag{1}$$

[1] Calculation of the sensitivity coefficients (as used for standard uncertainty evaluation of uncorrelated input quantities in Ref. [2]) is provided in Ref. [23] and applied in Ref. [24]. Following the calculations in Ref. [23], we assume a sign error for uncertainty from data timestamps in Ref. [24]. In this publication, we consider the version from Ref. [23].

$$u_{\hat{y}}^2(t) = \underbrace{\sum_{i=1}^{N} F_i^2(t)u^2(y_i)}_{\text{unc. from data values}} + \underbrace{\sum_{i=1}^{N} F_i^2(t)\left(\left.\frac{\partial \hat{y}}{\partial t}\right|_{t=t_i}\right)^2 u^2(t_i)}_{\text{unc. from data timestamps}} + \underbrace{\left(\frac{\partial \hat{y}}{\partial t}\right)^2 u^2(t)}_{\text{unc. of requested time}}, \quad (2)$$

where (t_i, y_i) denotes the original data points and F_i the interpolation kernels. Note, that the last two terms in the uncertainty equation are zero if both time uncertainty $u(t_i)$ and uncertainty of the requested time $u(t)$ are negligible and therefore set to zero.

The interpolation method is not fully online-capable, but can be performed iteratively, allowing for an execution before all data is recorded.

4.5. Uncertainty from dynamic calibration

An established dynamic calibration method using sine excitation is performed [11] but under consideration of absolute timestamps and using digital processing as opposed to analog phase synchronization. In order to perform the calibration, PTB's three component acceleration facility was used to excite the *device under test* (DUT), i.e. the sensors shown in Fig. 6. The acceleration sensors are excited with monofrequent sine signals at different frequencies (10 Hz–200 Hz in steps of 10 Hz) and amplitudes (between 12 m/s^2 to 50 m/s^2) along all three measurement axes (X, Y, Z) in the laboratory reference frame. The mechanical excitations are measured with three *laser Doppler vibrometers* (LDVs) as references. A sine approximation is fitted to the LDV velocity values, transferring it to the frequency space. By derivation of the velocity in the frequency domain, this leads to the actual acceleration values. Since the sensor coordinate frames do not perfectly match the laboratory reference frame, the rotation angles must be determined. For this purpose, signed amplitude vectors are calculated from the amplitude and phase values of the frequencies up to 40 Hz, taking into account the group delay. The rotation matrix for each sensor can be determined using the *Kabsch algorithm* [25,26] and is implemented using the SciPy function `align_vectors`. Based on [11], the complex frequency response values are calculated from the time synchronized LDV and the sensor readings.

A stable *infinite impulse response* (IIR) filter is chosen to represent the inverse transfer behavior. This is achieved by a least square fit (LSIIR) to the reciprocal of a given set of frequency response values and their corresponding uncertainties. Only raw data points that are reasonably different from zero for both DUT and reference system are used for the transfer behavior estimation to focus on the transfer characteristics along the same axis of both systems. The implementation makes use of the `PyDynamic` function `invLSIIR_unc` as described in Ref. [27], which propagates uncertainties according to the *Guide to the Expression of Uncertainty in Measurement Supplement 2* (GUM S2) Monte Carlo method [28]. This leads to the filter numerator and denominator coefficients $(\vec{b}, \vec{a})$, the time delay τ in samples and the uncertainties associated with the filter coefficients.To evaluate dynamic uncertainty of the sensor data, the obtained IIR filter with uncertainty is applied to the raw sensor readings, yielding a compensated signal with dynamic measurement uncertainties based on the calibration results. This is accomplished using the `PyDynamic` function `IIRuncFilter` which is based on the formulas given in Ref. [29]. The amplitude spectrum of the empirical transfer behavior in DUT-y-direction and the fitted inverse behavior are visualized in Fig. 9 with coefficients

$$\vec{b} \approx [0.54, -0.59, 0.58, -0.19] \text{ and} \quad (3)$$

$$\vec{a} \approx [1.0, -2.11, 3.15, -3.10, 2.20, -1.09, 0.30]. \quad (4)$$

Applying the inverse behavior to 1 s of the 51st cycle yields the time series plot in Fig. 10.

5. Lifetime estimation with ML

After performing the alignment of the ZeMA DAQ and the SUU data, the influence of multiple interpolation schemes on subsequent ML pipelines and on the associated processing steps is investigated in this section.

5.1. Automated ML toolbox

To evaluate the data sets, a software toolbox for statistical ML [19, 20,30] is used. An uncertainty-aware version of this toolbox has been developed recently,[2] but was not ready for this publication. The automated ML toolbox is particularly suitable for analysis of cyclic sensor data and consists of three main parts (cf. Fig. 11): *feature extraction*, *feature selection*, and *classification*. For feature extraction, five complementary methods are used which extract features from the time, frequency and time-frequency domains. Feature selection is carried out with three complementary methods. In this step, redundant features and features with low information content are removed from the feature set. Feature extraction together with feature selection leads to 15 possible algorithm combinations. Classification is further split in two parts: an additional dimensionality reduction step using *Linear Discriminant Analysis* (LDA) and the classification itself which is based on the Mahalanobis distance. To validate the results, a 10-fold stratified cross-validation is used. This means, the data set is equally partitioned into ten subsets and the class distribution within the subsets is nearly equal. For every fold, the model is trained with the training data (90% of the data set) and the resulting model is then applied to the test data (10% of the data set). For every fold, the cross-validation error, i.e. the percentage of misclassified cycles, is calculated and averaged over all folds. The algorithm combination with the lowest cross-validation error out of the 15 combinations is chosen as the best for the classification task at hand.

5.2. Results and interpretation

The automated ML toolbox is applied to the preprocessed data of both measurement systems with the target classification given by the

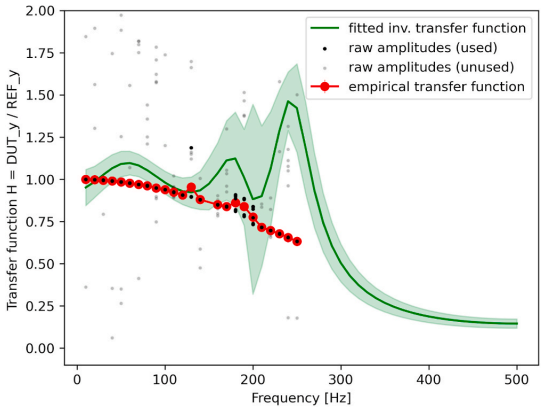


Fig. 9. Amplitude spectrum of empirical transfer behavior in DUT-y-direction (red) and fitted inverse transfer function (green). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

[2] https://github.com/ZeMA-gGmbH/LMT-UA-ML-Toolbox

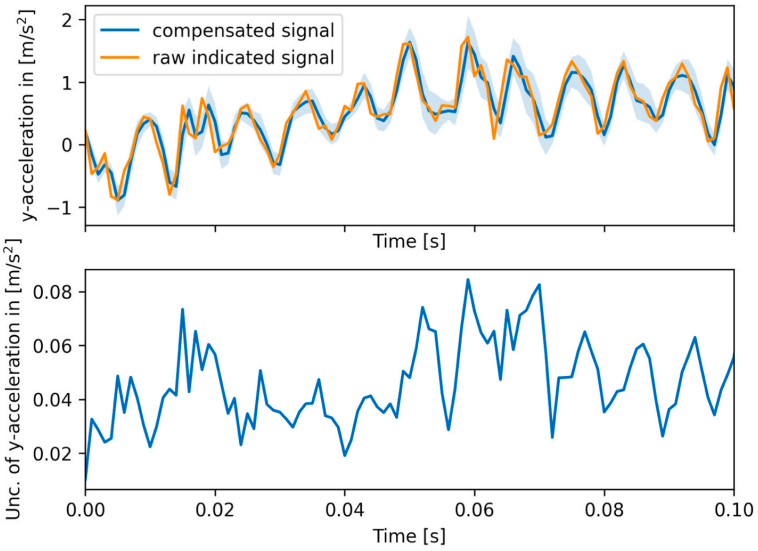


Fig. 10. Top: Comparison of indicated and compensated time series of the *y*-axis of the DUT (note: for better visualization an expanded uncertainty with $k = 5$ is shown). Bottom: Corresponding dynamic standard uncertainty of the compensated signal ($k = 1$).

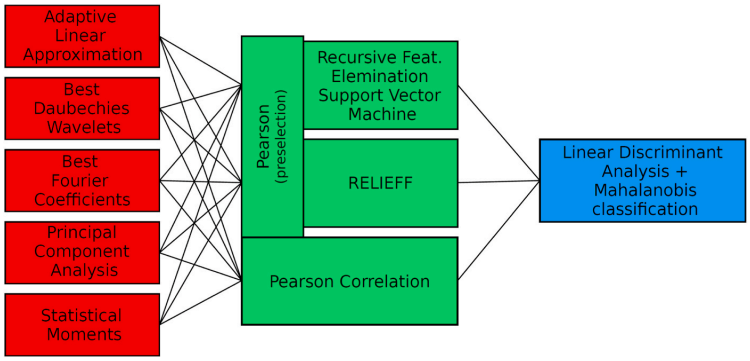


Fig. 11. Scheme of the ML toolbox with feature extraction (red), selection (green) and classification (blue) [30]. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

percentage of lifetime already passed by starting at 1%. The best result, i.e. the smallest cross-validation error, is achieved with *Best Fourier Coefficients* (BFC) as extractor and *Recursive Feature Elimination Support Vector Machine* (RFESVM) as selector. In Fig. 12, the influence of the measurement system and the interpolation/resampling method can be clearly seen. More precise and accurate lifetime predictions are achieved with the ZeMA DAQ data for 1 kHz and 2 kHz sampling rate. As seen in Fig. 12 the effect of the chosen interpolation scheme has an influence on the ML training. While `nearest` (nearest neighbor), `linear` and `cubic` perform similarly well, the interpolation methods `next` and `previous` show a decrease in performance of approx. 10%.

The classification error for 1% ($\widehat{=}$ 3.88 h) lifetime target increments is 26.47% for the ZeMA DAQ data in comparison to 45.95% for the cubic interpolated SUU data as shown in Fig. 13. The root-mean-square error (RMSE) for 1% lifetime target increments is 1.39% for the ZeMA DAQ data in comparison to 5.25% for the cubic interpolated SUU data.

However, reducing the required accuracy in the lifetime target to 10% ($\widehat{=}$ 38.78 h) improves the prediction quality for the SUU data and leads to usable classification errors of 12% as shown in Fig. 14. The larger lifetime target increments together with the low cost of the SUU hardware results in a good tradeoff between cost and accuracy for many use cases.

Fig. 15 shows which individual sensors from the SUU data actually contributed into the ML lifetime estimation. The acceleration sensors provide 90% of all features (19 in total) used for the ML model building.

Repeating the lifetime estimation using only one of the sensors installed at the plain bearing (Kistler 8712A5M1, MPU 9250 or BMA 280), yields very similar cross-validation errors of 67.33%, 63.89% and 65.88%, respectively. This allows to conclude that the more accurate lifetime estimation of the ZeMA DAQ system is not so much a cause of the better acquisition performance (larger sampling rate and resolution, high time accuracy, high-end sensors in comparison to the sensors of the SUU), but rather a consequence of the available variety of measurands.

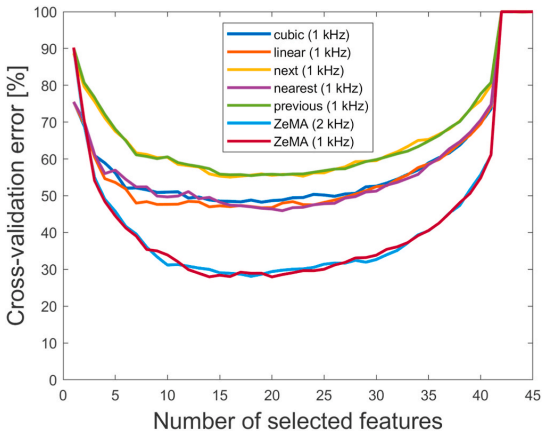


Fig. 12. Cross-validation performance of models trained on different input data sets. BFC is used as extractor and RFESVM as selector.

6. Conclusion and outlook

Driven by the idea of bringing together data from two different data acquisition units (DAQs), data of one DAQ was brought into the event-centric format of the other. The preprocessing step necessary to achieve this turned out to be very computationally complex and time-consuming. Therefore, careful data pre-inspection is necessary and solutions to compensate the encountered problems are given. These solutions could lead to enhancements that reduce the required effort for data preprocessing in future measurement campaigns.

Uncertainty information for the SUU data is obtained from dynamic calibration and corresponding compensation with an uncertain filter. The influence of timestamp uncertainty was investigated, but its overall contribution to the uncertainty of the interpolated signal is minor because of the use of absolute timestamps provided by the onboard GNSS module. However, this can change drastically, if the provided time-signal has significantly higher uncertainty.

The interpolated SUU data clearly indicates that a sensing system of much lower cost can also provide raw data suitable for an ML lifetime estimation. However, this necessitates involved preprocessing in conjunction with the fact that the estimated lifetime from the SUU data does not achieve the same resolution for the remaining useful lifetime prediction as the more complex sensor system represented by the ZeMA data (10% vs. 1% lifetime target increments). But following the idea of "fitness for purpose", this is still an excellent example for the adequacy of the measurement effort and required accuracy. As these lifetime estimations will be used as an indicator in predictive maintenance, 10% increments will likely be sufficient for most maintenance interval scheduling tasks.

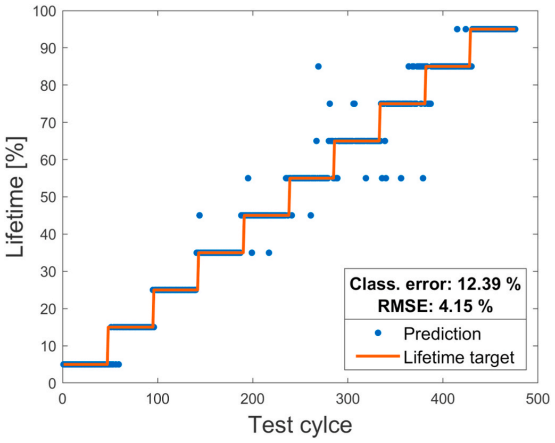


Fig. 14. Predicated lifetime of cubic interpolated SUU data compared with linear target (10% increments) for one fold of the 10-fold cross-validation.

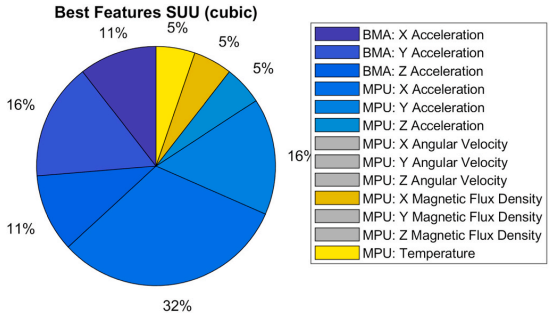


Fig. 15. Percentage of features selected from different sensors on the SUU that contribute to the lifetime estimate.

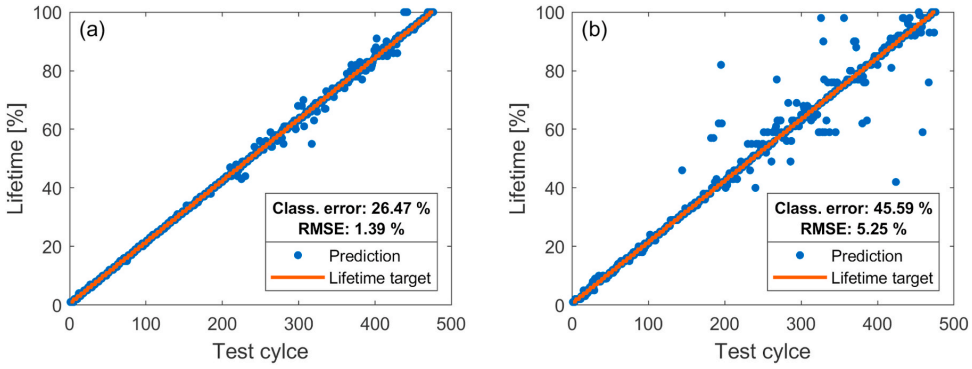


Fig. 13. Predicated lifetime compared with linear target (1% increments) for one fold of the 10-fold cross-validation. (a) 1 kHz ZeMA DAQ data and (b) 1 kHz cubic interpolated SUU data.

In an upcoming measurement campaign, the SUU data will be sampled at 2 kHz. This would also allow extraction of features from the 500 Hz–1000 Hz range, which is otherwise not possible as the used 1 kHz sampling rate is below the required Nyquist-rate of 2 kHz. This is necessary, because highly relevant features from the original ZeMA data are known to be in the range of 0 Hz–1000 Hz [1]. The transferability of an abstraction of the model generated with data of one EMC to another EMC is an ongoing research topic at ZeMA, the current focus lies on domain adaption methods [31]. As the uncertainty values are not used for the ML model building, the application of a recently developed uncertainty-aware automated ML toolbox will be investigated in upcoming research.

CRediT authorship contribution statement

Tanja Dorst: Conceptualization, Methodology, Software, Data curation, Formal analysis, Investigation, Writing – original draft. **Maximilian Gruber:** Conceptualization, Methodology, Software, Data curation, Formal analysis, Writing – original draft. **Benedikt Seeger:** Software, Investigation, Resources. **Anupam Prasad Vedurmudi:** Writing – review & editing. **Tizian Schneider:** Software, Resources. **Sascha Eichstädt:** Writing – review & editing, Supervision. **Andreas Schütze:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

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 automated 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.

We acknowledge support by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) and Saarland University within the funding programme Open Access Publishing.

Appendix A. On Time Uncertainty in Interpolation

As described in a recent publication [1], time uncertainty or shifts/drifts can have a significant effect on the performance of subsequent ML-processes. A small yet accumulating deviation in the phase between the raw and interpolated data points over a period of 30 ms is observable in the lowermost plot in Fig. 8. Under the assumption that the raw data is recorded at its nominal sample rate (1000 Hz), the end of the cycle ($t_0 + 2.8$ s) would be after 2800 data points. However, knowing the absolute timestamps, the 2800th datapoint in the raw data corresponds (in cycle 51) to just $t_0 + 2.7889$ s. Over just one cycle the timestamp error would have already accumulated to 11.11 ms (or 11 data points respectively). As described in Ref. [1], this can lead to a reduced prediction performance of trained ML-methods if left untreated.

Such time uncertainty can be incorporated into the uncertainty analysis of the interpolated value. The time information coming from the SUU is based on GNSS and typically achieves time uncertainty of around ~ 300 ns. The implemented interpolation method presented above does not consider time uncertainty information, although equation (2) supports it. Therefore the influence of time uncertainty is manually compared for some interesting cases by evaluating the uncertainty from the data timestamps (second term in equation (2))

$$\sum_{i=1}^{N} F_i^2(t) \left(\left. \frac{\partial \hat{y}}{\partial t} \right|_{t=t_i} \right)^2 u^2(t_i) \tag{A.1}$$

and compared to the uncertainty from the data values (first term in equation (2))

$$\sum_{i=1}^{N} F_i^2(t) u^2(y_i). \tag{A.2}$$

In the following, it is assumed that the uncertainty of the requested time (third term in equation (2)) is zero. For the 51st cycle the median of the root of the first term (equation A.2) over all data points evaluates to 2.236×10^{-3} m/s^2. At the expected time uncertainty achievable in GNSS based systems the median of the root of the second term evaluates to approximately 1×10^{-4} m/s^2. The variation in the median of the root of equation A.1) is shown in Fig. 16 for different values of given input time uncertainties. Specific indicators have been placed on values corresponding to many relevant magnitudes found in practical applications such as:

- GNSS[3]: ~300 ns
- Precision Time Protocol IEEE 1588: ~300 ns [32].
- local Network Time Protocol (NTP): ~100 µs [33].
- Trigger detection[4]: ~1 ms
- web NTP: ~10 ms [33].

As can be seen in Fig. 16, the observed uncertainty of ~300 ns for the GNSS timestamps (yellow line) would not contribute much to the uncertainty of the interpolated value. For other above mentioned common time sources this behavior changes.

[3] as indicated by our data using the onboard oscillator of the SUU's debugger.
[4] in the setup of this paper.

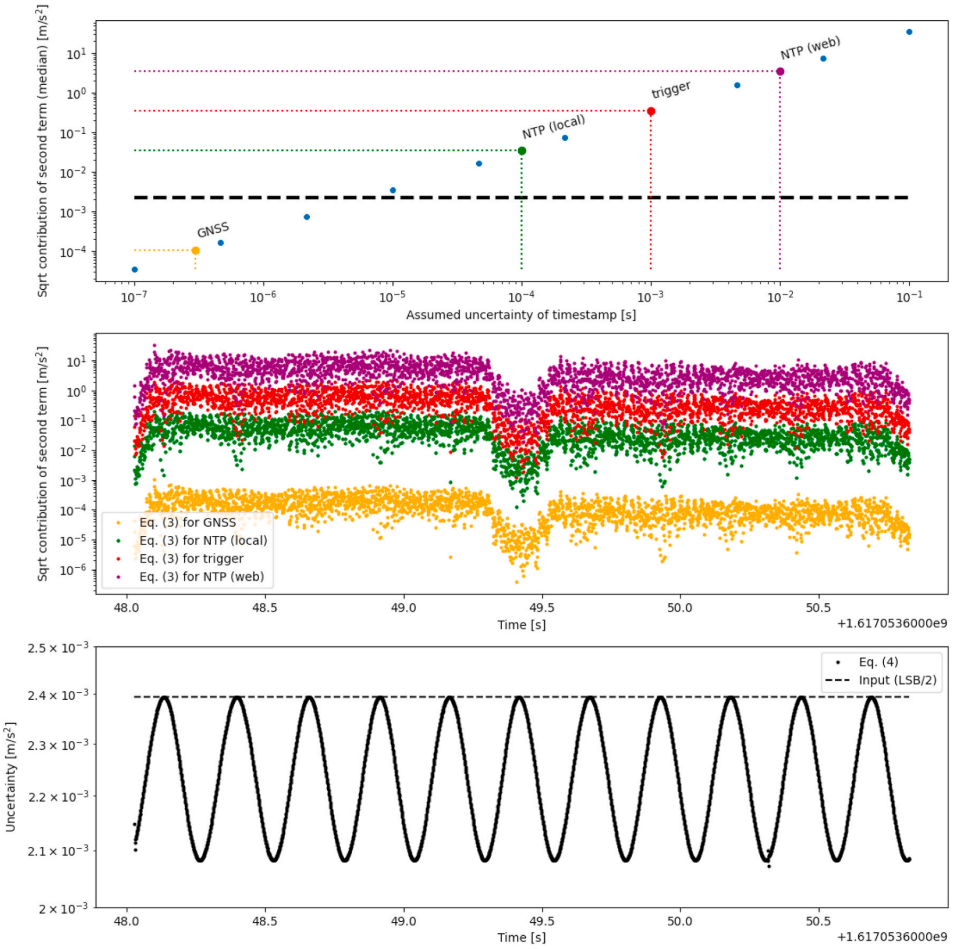


Fig. 16. Contribution of timestamp uncertainty onto interpolated data values for different assumed input time uncertainties.

Appendix B. Exemplary Raw Data

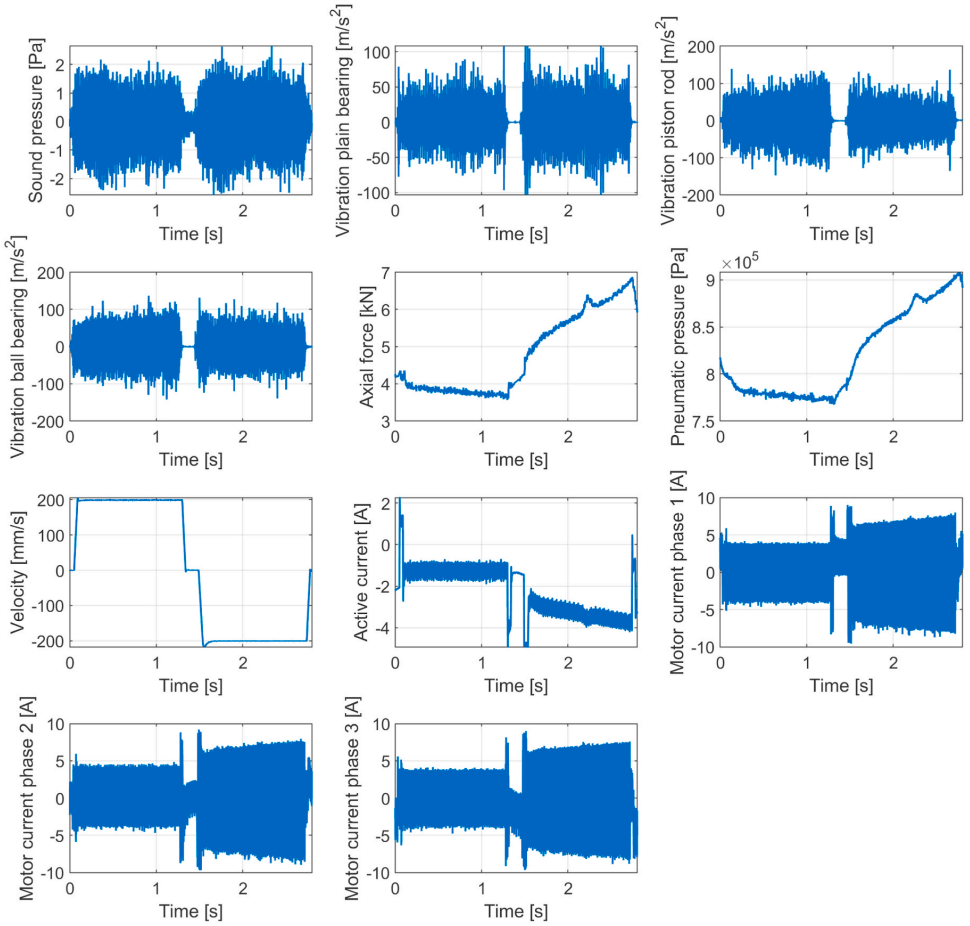


Fig. 17. Raw data recorded during the 51st cycle by the ZeMA DAQ expressed in SI units.

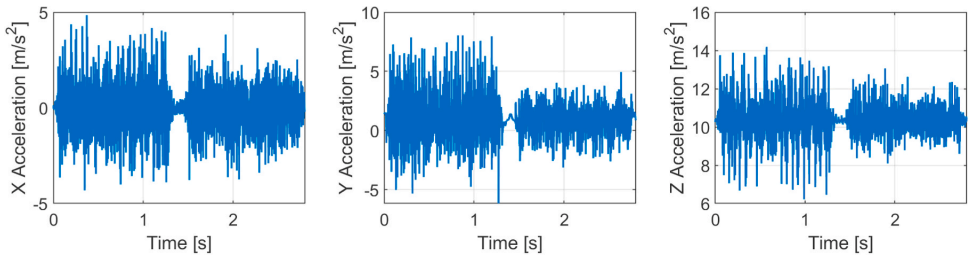


Fig. 18. Raw data recorded during the 51st cycle by the BMA 280 sensor of the SUU.

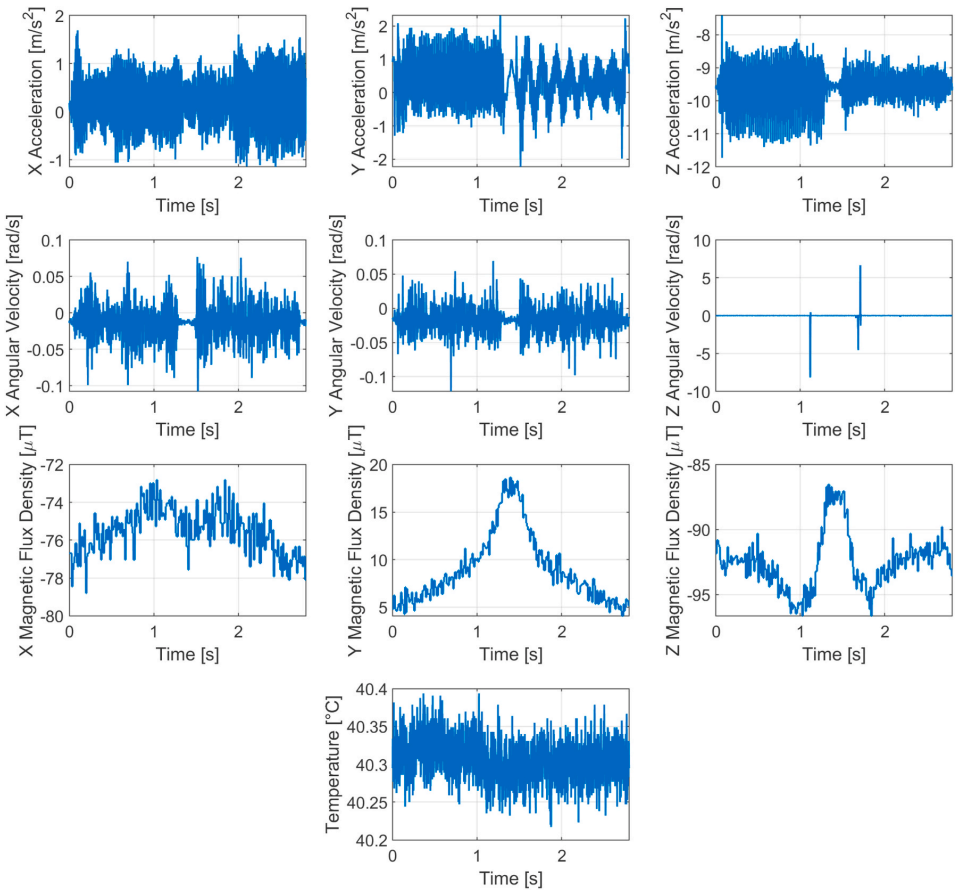


Fig. 19. Raw data recorded during the 51st cycle by the MPU 9250 sensor of the SUU.

References

[1] T. Dorst, Y. Robin, S. Eichstädt, A. Schütze, T. Schneider, Influence of synchronization within a sensor network on machine learning results, J. Sens. Sens. Syst. 10 (2) (2021) 233–245, https://doi.org/10.5194/jsss-10-233-2021.

[2] BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP, OIML, Guide to the Expression of Uncertainty in Measurement, 2008.

[3] BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP, OIML, International Vocabulary of Metrology - Basic and General Concepts and Associated Terms, VIM, 2012.

[4] S.E. Festo, K.G. Co, Electric cylinders ESBF, with spindle drive. https://www.festo.com/cat/en-gb_gb/data/doc_ENUS/PDF/US/ESBF_ENUS.PDF. (Accessed 19 April 2022).

[5] N. Helwig, Zustandsbewertung industrieller Prozesse mittels multivariater Sensordatenanalyse am Beispiel hydraulischer und elektromechanischer Antriebssysteme, PhD thesis, Saarland University, Dept. Systems Engineering, 2018.

[6] N. Helwig, T. Schneider, A. Schütze, MoSeS-Pro: Modular sensor systems for real time process control and smart condition monitoring using XMR-technology, Proc. 14th Symposium Magnetoresistive Sensors and Magnetic Systems.

[7] National Instruments, PXIe-5170. https://www.ni.com/en-us/support/model.pxie-5170.html. (Accessed 19 April 2022).

[8] National Instruments, PXIe-4492. https://www.ni.com/en-us/support/model.pxie-4492.html. (Accessed 19 April 2022).

[9] National Instruments, PXIe-6341. https://www.ni.com/en-us/support/model.pxie-6341.html. (Accessed 19 April 2022).

[10] STMicroelectronics, STM32F767ZI, in: https://www.st.com/en/microcontrollers-microprocessors/stm32f767zi.html. (Accessed 19 April 2022).

[11] B. Seeger, T. Bruns, Primary calibration of mechanical sensors with digital output for dynamic applications, ACTA IMEKO 10 (2021) 177–184, https://doi.org/10.21014/acta_imeko.v10i3.1075.

[12] D. Hutzschenreuter, F. Härtig, W. Heeren, T. Wiedenhöfer, A. Forbes, C. Brown, I. Smith, S. Rhodes, I. Linkeová, J. Sýkora, V. Zelený, B. Ačko, R. Klobučar, P. Nikander, T. Elo, T. Mustapää, P. Kuosmanen, O. Maennel, K. Hovhannisyan, B. Müller, L. Heindorf, V. Paciello, SmartCom Digital System of Units (D-SI) Guide for the use of the metadata-format used in metrology for the easy-to-use, safe, harmonised and unambiguous digital transfer of metrological data, doi:10.5281/zenodo.3522631.

[13] S. Eichstädt, Publishable Summary for 17IND12 Met4FoF "Metrology for the Factory of the Future", Nov. 2020, https://doi.org/10.5281/zenodo.4267955.

[14] A. Schütze, N. Helwig, T. Schneider, Sensors 4.0 – smart sensors and measurement technology enable industry 4.0, J. Sens. Sens. Syst. 7 (1) (2018) 359–371, https://doi.org/10.5194/jsss-7-359-2018.

[15] InvenSense Inc., MPU-9250 Product Specification. https://invensense.tdk.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf, 2016. (Accessed 19 April 2022).

[16] Bosch Sensortec GmbH, Data sheet BMA280 - Digital, triaxial acceleration sensor. https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bma280-ds000.pdf, 2021. (Accessed 19 April 2022).

[17] TE Connectivity, MS5837-02BA. https://www.te.com/commerce/DocumentDelivery/DDEController?Action=showdoc&DocId=Data+Sheet%7FMS5837-02BA01%7FA8%7Fpdf%7FEnglish%7FENG_DS_MS5837-02BA01_A8.pdf%7FCAT-BLPS0059, 2019. (Accessed 19 April 2022).

[18] Kistler Instrument Corporation, K-Shear®Accelerometer - Type 8712A5M1. https://intertechnology.com/Kistler/pdfs/Accelerometer_Model_8712A5M1.pdf, 2008. (Accessed 19 April 2022).

[19] T. Schneider, N. Helwig, A. Schütze, Automatic feature extraction and selection for classification of cyclical time series data, TM - Tech. Mess. 84 (3) (2017) 198–206, https://doi.org/10.1515/teme-2016-0072.
[20] T. Schneider, N. Helwig, A. Schütze, Industrial condition monitoring with smart sensors using automated feature extraction and selection, Meas. Sci. Technol. 29 (9) (2018), 094002, https://doi.org/10.1088/1361-6501/aad1d4.
[21] T. Dorst, M. Gruber, A.P. Vedurmudi, Sensor Data Set of One Electromechanical Cylinder at ZeMA Testbed (ZeMA DAQ and Smart-Up Unit), sep 2021, https://doi.org/10.5281/zenodo.5185953.
[22] A.F. Siegel, Robust regression using repeated medians, Biometrika 69 (1) (1982) 242–244, https://doi.org/10.1093/biomet/69.1.242.
[23] D.R. White, P. Saunders, The propagation of uncertainty with calibration equations 18 (7) (2007) 2157–2169, https://doi.org/10.1088/0957-0233/18/7/047.
[24] D.R. White, Propagation of uncertainty and comparison of interpolation Schemes, Int. J. Thermophys. 38 (3) (2017) 39, https://doi.org/10.1007/s10765-016-2174-6.
[25] W. Kabsch, A solution for the best rotation to relate two sets of vectors, Acta Crystallogr. A 32 (5) (1976) 922–923, https://doi.org/10.1107/S0567739476001873.
[26] W. Kabsch, A discussion of the solution for the best rotation to relate two sets of vectors, Acta Crystallogr. A 34 (5) (1978) 827–828, https://doi.org/10.1107/S0567739478001680.
[27] S. Eichstädt, C. Elster, T.J. Esward, J.P. Hessling, Deconvolution filters for the analysis of dynamic measurement processes: a tutorial, Metrologia 47 (5) (2010) 522–533, https://doi.org/10.1088/0026-1394/47/5/003.
[28] BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP, OIML, Supplement 2 to the "Guide to the Expression of Uncertainty in Measurement" – Extension to Any Number of Output Quantities, 2011.
[29] A. Link, C. Elster, Uncertainty evaluation for IIR (infinite impulse response) filtering using a state-space approach, Meas. Sci. Technol. 20 (5). doi:10.1088/0957-0233/20/5/055104.
[30] T. Dorst, Y. Robin, T. Schneider, A. Schütze, Automated ML Toolbox for Cyclic Sensor Data, Mathematical and Statistical Methods for Metrology (MSMM).
[31] P. Goodarzi, A. Schütze, T. Schneider, Comparison of different ML methods concerning prediction quality, domain adaptation and robustness, TM - Tech. Mess. 89 (4) (2022) 224–239, https://doi.org/10.1515/teme-2021-0129.
[32] S. Schriegel, D. Kirschberger, H. Trsek, Reproducible IEEE 1588-performance tests with emulated environmental influences, in: 2010 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication, 2010, pp. 146–150, https://doi.org/10.1109/ISPCS.2010.5609783.
[33] D.L. Mills, J. Martin, J. Burbank, W. Kasch, Network time Protocol version 4: Protocol and algorithms Specification, Tech. Rep. (Jun. 2010), https://doi.org/10.17487/RFC5905.

3.3 Making a data set FAIR

FAIRness and metrological traceability build a basis for measurement data exchange in research and industry. As an extension to Paper 1, the EMC lifetime test data sets of both DAQ systems, the ZeMA DAQ and the SUU, used in this paper were not only uploaded on Zenodo as an HDF5 file but also made FAIR beforehand to ensure sustainable research data management. To achieve FAIRness, the data sets were restructured, merged, and extended with metadata that fits the FAIR principles explained in Section 2.4 [213, 214]. For the metadata, several ontologies and knowledge representations are used.

The top-level metadata contains information about the data set, e.g., information about the creators of the data set and their contact addresses, publication-related information, or information about the experiment carried out to acquire the data. Listing 3.1 shows the most important top-level metadata in the *JavaScript Object Notation* (JSON) format. A complete list of all top-level metadata for this data set can be found on Zenodo [215].

```
"Project": {
   "fullTitle": "Metrology for the Factory of the Future",
   "funding programme": "EMPIR",
   "fundingNumber": "17IND12"
},
"Person": {
   "dc:author": ["Tanja Dorst", "Maximilian Gruber", "Anupam Prasad
       Vedurmudi"],
   "e-mail": ["t.dorst@zema.de", "maximilian.gruber@ptb.de","anupam
      .vedurmudi@ptb.de"],
   "affiliation": ["ZeMA gGmbH", "Physikalisch-Technische
      Bundesanstalt", "Physikalisch-Technische Bundesanstalt"]
},
"Publication": {
   "dc:identifier": "10.5281/zenodo.5185953",
   "dc:license": "Creative Commons Attribution 4.0 International (
      CC-BY-4.0)",
   "dc:title": "Sensor data set of one electromechanical cylinder
      at ZeMA testbed (ZeMA DAQ and Smart-Up Unit)",
   "dc:subject": ["measurement uncertainty", "sensor network", "
      MEMS"],
   "dc:SizeOrDuration": "24 sensors, 4776 cycles and 2000
      datapoints each"
```

```
},
"Experiment": {
   "date": "2021-03-29/2021-04-15",
   "DUT": "Festo ESBF cylinder",
   "identifier": "axis11"
}
```

Listing 3.1: JSON code for the most important top-level metadata [213, 214].

As ZeMA DAQ and SUU data are merged in one data set, the HDF5 file is structured in two main groups, one for each system. The subgroups of these two main groups contain the numerical measurement values together with machine-readable descriptions of the corresponding sensors, quantities, and units. They also contain important information, like whether the values are ADC or converted values or if an interpolation scheme has been used to obtain the numerical measurement values. Listing 3.2 shows two JSON code examples, one for a sensor of the ZeMA DAQ and one for a sensor of the SUU.

```
"/ZeMA_DAQ/Sound_Pressure": {
   "sosa:madeBySensor": "G.R.A.S.46BE",
   "rdf:type": "qudt:Quantity",
   "si:unit": "\\pascal",
   "qudt:hasQuantityKind": "qudt:SoundPressure",
   "qudt:value": {
      "si:label": "Sound pressure",
      "misc": {
         "raw_data": False,
         "comment": "Converted from ADC values based on appropriate
             conversion."
      },
   },
   "qudt:standardUncertainty": {
   "si:label": "Sound pressure uncertainty"
   }
},
"/PTB_SUU/BMA_280/Acceleration": {
   "sosa:madeBySensor": "BMA 280",
   "rdf:type": "qudt:Quantity",
   "si:unit": "\\metre\\second\\tothe{-2}",
   "qudt:hasQuantityKind": ["qudt:Acceleration", "qudt:Acceleration
      ", "qudt:Acceleration"],
   "qudt:value": {
      "si:label": ["X acceleration", "Y acceleration", "Z
         acceleration"]
```

```
24      },
25      "qudt:standardUncertainty": {
26          "si:label": ["X acceleration uncertainty", "Y acceleration
                uncertainty", "Z acceleration uncertainty"]
27      },
28      "misc": {"interpolation_scheme": "cubic"}
29  }
```

Listing 3.2: JSON code for the metadata of the sound pressure sensor G.R.A.S.46 BE of the ZeMA DAQ and the 3-axis accelerometer BMA 280 of the SUU [213, 214].

To assess the FAIRness level of the merged data set [215], the FAIR data maturity model is applied. Figure 3.2 shows a detailed assessment of the 41 indicators, divided into the four main FAIR data principles. Evaluating all indicators shows a good agreement between the data set and the FAIR data principles. Nevertheless, certain indicators should still be further improved. However, even if the maximum value (4) for all indicators is reached, this is not a guarantee to represent meaningful data.

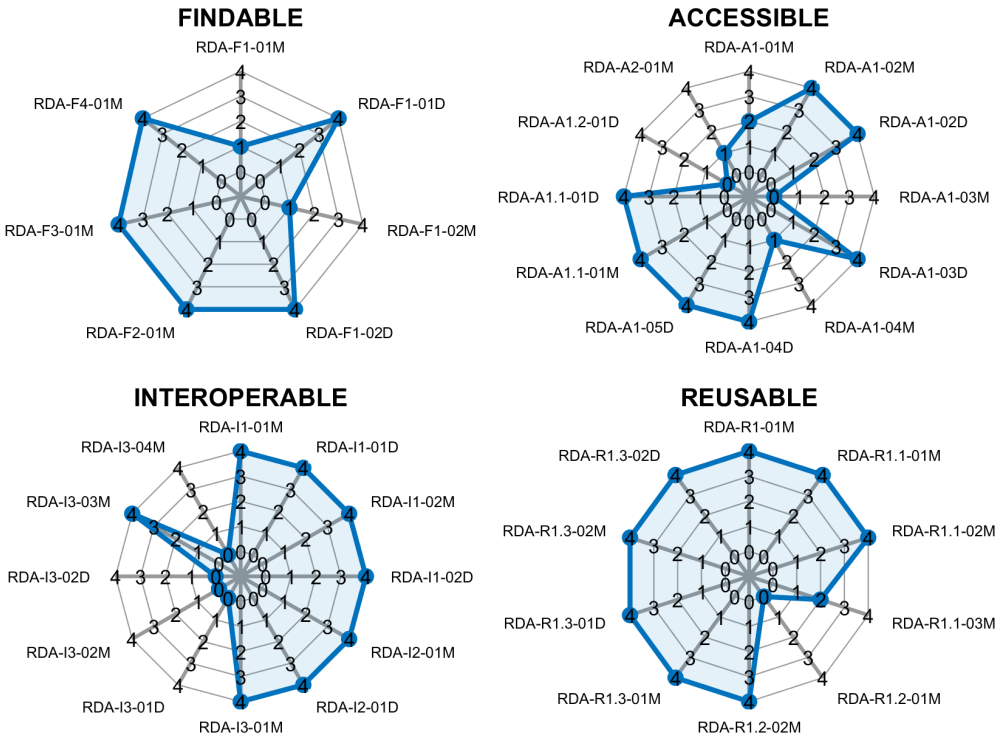


Figure 3.2: Detailed assessment of the indicators of the FAIR data maturity model (adapted from [214]).

3.4 Paper 2 – Influence of synchronization within a sensor network on machine learning results

Digitalization in the context of I4.0 enhances strategic flexibility leading to more productive and efficient processes as well as improving product quality [9]. In a Factory of the Future, the IIoT builds the networking basis for enabling interconnected sensors, machines, and other devices, so that they are able to communicate with each other. Sensor networks are a key component in the IIoT, allowing data collection and data-driven analysis [200]. Smart sensors and data evaluation, e.g., based on ML, allows, inter alia, fault condition detection [37], predictive maintenance [216], as well as production planning and control [217]. In order to fully use the potential of smart sensors, it is essential to consider the quality of the sensor data [196]. The data quality is influenced, e.g., by environmental conditions, sensor precision, sensor failure, or time synchronization problems. As sensors in distributed sensor networks can sample at different times, data of these sensors will not have the same time base, i.e., time synchronization errors between multiple sensors can occur. Additionally, time delays can occur when data is transmitted within a sensor network [218]. For ML applications, correctly performed data fusion is crucial, as shown in the presented paper.

Paper 2 [219] addresses the data quality problem through time synchronization errors occurring mainly within distributed sensor systems and investigates the influence of these time synchronization errors on ML results. For the presented study, an EMC data set acquired during a lifetime test, as described in Section 2.3.2, is used. ML is performed with only one second of the return stroke phase because the velocity and load are constant during this period. As the full data set of one lifetime test consists of approx. 629,000 cycles meaning approx. 12 TB, it is essential to decrease computational cost by reducing the amount of data used for ML with the AMLT. First, a cycle reduction is carried out using only every 100^{th} working cycle, leading to approximately 62 cycles per target class. A possibility to further reduce the amount of data is reducing the sampling rate of the data set. To test if lower sampling rates influence ML results, several data sets with various sampling rates are used for model training in the AMLT. It is shown that using the data set, which has been downsampled to 2 kHz, and only contains every 100^{th} working cycle, is sufficient to obtain similar results compared to the data set with full sampling rates for each sensor. Data sets with lower sampling rates perform worse than the 2 kHz data set, whereas data sets with higher sampling rates up to 10 kHz perform similarly to the 2 kHz data set. As most of the relevant

features for the data sets of different sampling rates are between 250 Hz and 1 kHz and the Nyquist criterion [220] requires a sampling rate at least twice the highest frequency, the data set downsampled to already 2 kHz contains these relevant features. Therefore, the data set with a sampling rate of 2 kHz containing only every 100^{th} working cycle is chosen for further investigations. This reduced data set recorded during the lifetime test of axis 3 is available in the EMC data set presented in Section 2.3.2. It is used to train a classification ML model to predict the RUL of the EMC with a resolution of 1 %. *Best Fourier Coefficients* (BFC) for feature extraction (FE) and *Recursive Feature Elimination Support Vector Machine* (RFESVM) for feature selection (FS) is determined as the best algorithm combination by the AMLT, reaching a cross-validation (CV) error of 18.18 % in model training, which is a good result for this task. In comparison, when performing ML with the full data set and, therefore, dealing with extremely high computational cost, the minimum reachable CV error is 8.9 % using a more complex model with 499 features [221]. The CV error of 18.18 % is achieved when using only 17 Fourier coefficient features for RUL estimation, whereby 12 of them represent amplitudes. One advantage of using ML instead of deep learning (DL) is the interpretability of the model. In this contribution, it is shown that the 17 most important features have a physical explanation based on [139].

To investigate how the time shifts in the data influence the model training results, ML models are trained with different time-shifted data sets. Random time shifts with a minimum of 0.1 ms up to a maximum of 50 ms between individual sensors' working cycles are used to simulate time synchronization errors. The time-shifted data sets are generated using the full data set and downsampling it to 2 kHz after artificially generated random time shifts between individual sensors' working cycles are added. It is demonstrated that the larger the maximum time shift in the data set is, the worse the CV error in the training. Minimal synchronization errors have only a small impact on the CV error. However, for the data set with time shifts up to 50 ms, a CV error of 29.97 % is reached, which is significantly worse than the CV error of the raw data set (18.18 %). Likely, the variance in the data increases by increasing random time shifts, and therefore, learning for the model becomes more difficult.

For one amplitude feature (120 Hz of the active current) of the 17 most relevant features, it is exemplarily shown that amplitude changes during the lifetime of the EMC. However, the amplitudes are nearly equal for the same cycle number of the raw data set and the data set with time shifts up to 50 ms. The robustness of the amplitudes against time shifts can be explained using the mathematical expression of the Fourier transform.

Time-shifting a signal leads to an unchanged amplitude spectrum, whereas the phase spectrum experiences a frequency-proportional (linear) phase shift. This effect is used later in the contribution to improve ML models.

To investigate the influence of time shifts on the prediction performance, the ML model is trained with the raw data set, and this trained model is then applied to the data sets with different random time shifts. This approach simulates synchronization problems within a sensor network. For training (including validation) and testing, an 80:20 split is used, as explained in Section 2.2.4. It is shown that the classification error (the percentage of misclassified cycles) determined using the test data set increases with increasing time shifts. Already random time shifts of up to 0.1 ms lead to a significant classification error increase. If no better synchronization is possible between the sensors, the phase features can be excluded from the feature set after FE to enhance the results significantly. For example, using the data set with up to 1 ms time shifts, the classification error is enhanced to 44.99 % using only amplitudes as features in the trained model, in contrast to a classification error of 95.87 % using phases and amplitudes as features. Training with not only the raw data set but also, in addition, with time-shifted data sets and then removing the phases out of the resulting feature set is successfully tested to improve the model further. To test this improving effect, on the one hand, only the raw data set is used for model training, and on the other hand, in addition to the raw data set, the two data sets with time shifts of up to 0.1 ms and 0.5 ms are used for model training. This leads to two trained models based on amplitude and phase features. By removing the phases from the feature sets of both trained models, leading to two additional models containing only amplitudes as features, there are a total of four models whose performances are compared. Testing of the four models is then carried out by using the data set with time shifts of up to 1 ms. Both models trained with the raw data performed worse than the two models trained with the three data sets. The best model with a classification error of 35.93 % is that one trained with the three data sets and removed phases from the feature set.

Another important issue besides synchronization within a sensor network is the choice of the correct time frame of the data set for performing ML. It is indicated that the choice of the time frame for the 1 s period of the return stroke of the EMC is essential. To show this, training the model with the raw data set and applying it to different constant time-shifted data sets is carried out. Even in this case, enhancing the results by excluding the phase features from the feature set is possible.

In summary, Paper 2 provides suggestions for enhancing the setup of distributed measurement systems, especially concerning the necessary synchronization between sensors. When no information on the synchronization within the sensor network is available, artificially time-shifted data sets from the raw data set should be generated and used for model training.

J. Sens. Sens. Syst., 10, 233–245, 2021
https://doi.org/10.5194/jsss-10-233-2021

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-

Published by Copernicus Publications on behalf of the AMA Association for Sensor Technology.

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).

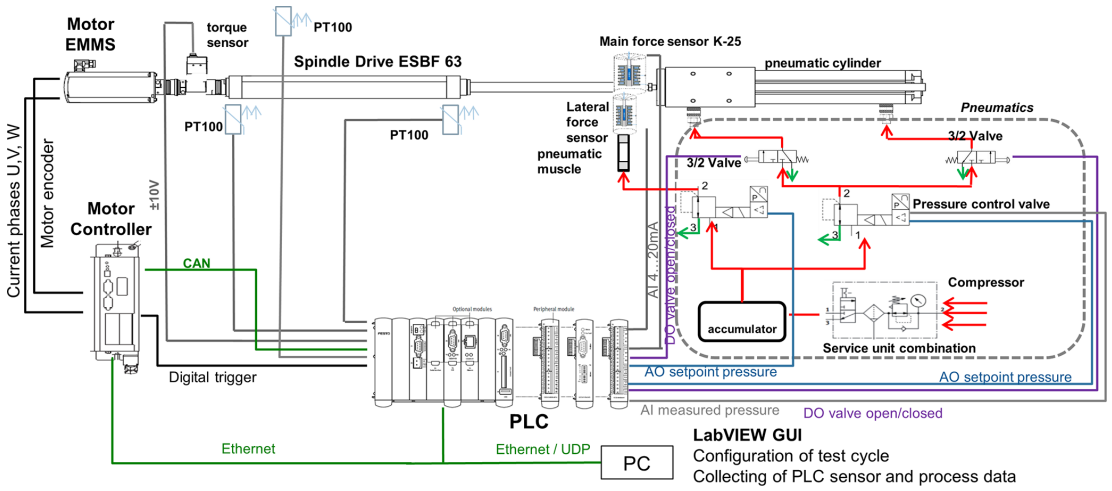


Figure 1. Basic scheme of the EMC test bed (Helwig et al., 2017).

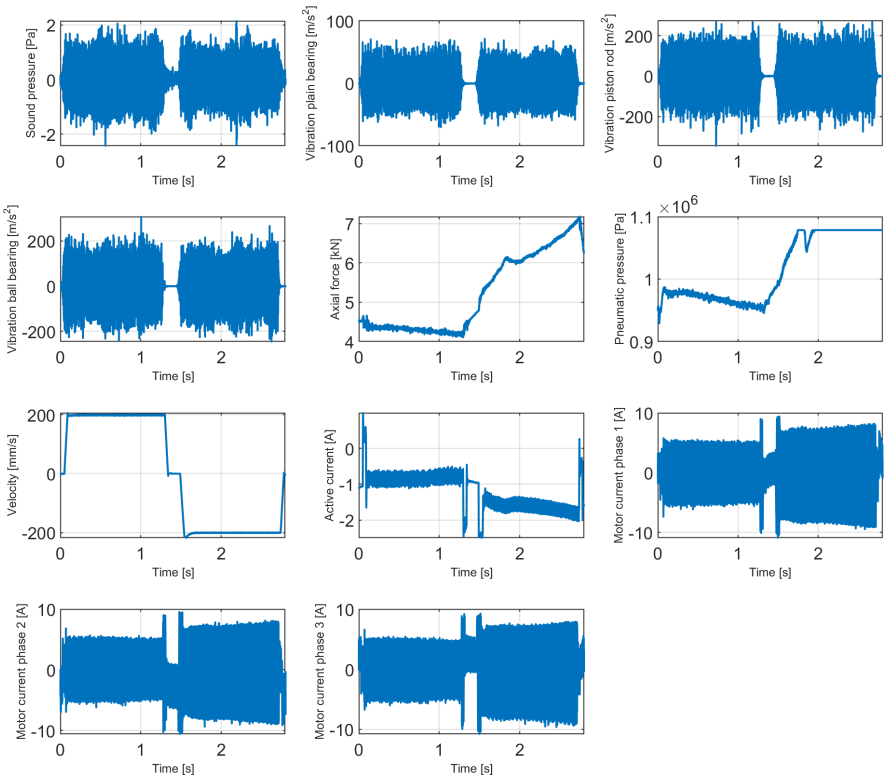


Figure 2. Raw data recorded during one cycle by 11 sensors expressed in SI units.

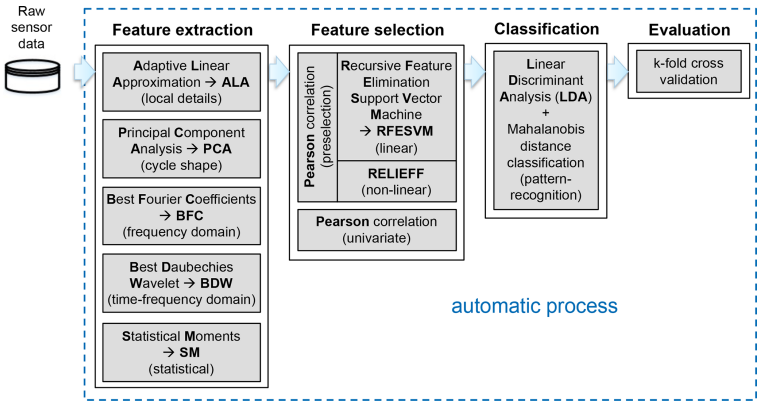


Figure 3. Schematic of the automatic toolbox for condition monitoring using machine learning, adapted from Dorst et al. (2019).

- 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.

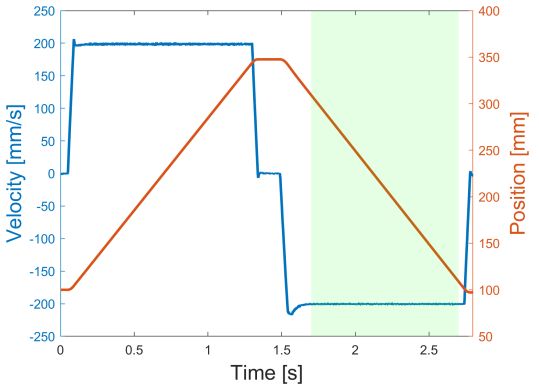


Figure 4. Working cycle depicted as position (red) and velocity (blue) consisting of forward stroke, waiting time, and return stroke, as well as the period (green) evaluated for estimation of the RUL.

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

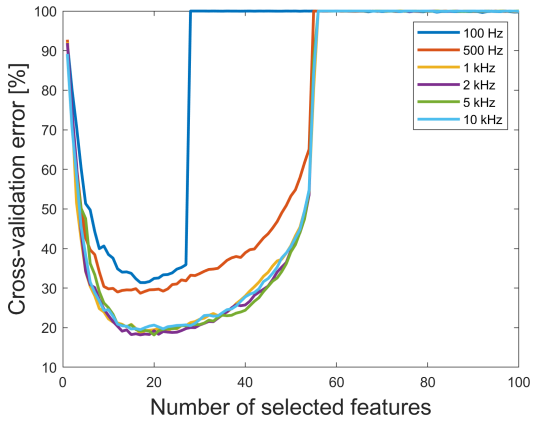


Figure 5. The 10-fold cross-validation error vs. number of selected features for data sets with different sampling rate using BFC as extractor and RFESVM as selector.

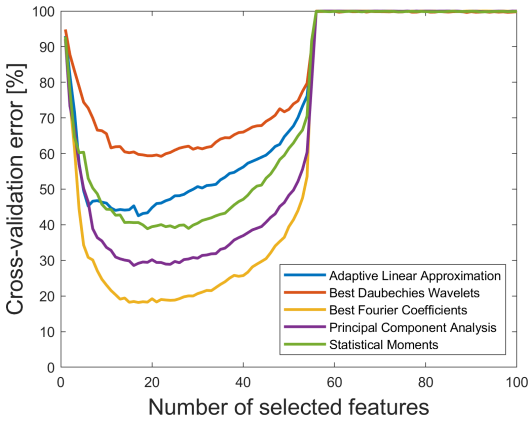


Figure 6. The 10-fold cross-validation error vs. number of selected features for the original 2 kHz data set without time shift using RFESVM as selector. For a better visibility, only results with RFESVM as selector are shown.

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

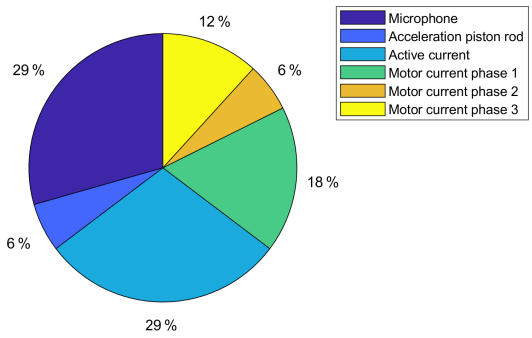


Figure 7. The 17 most important features by sensors, selected with RFESVM. Only 6 of the 11 sensors contribute to the 17 most important features.

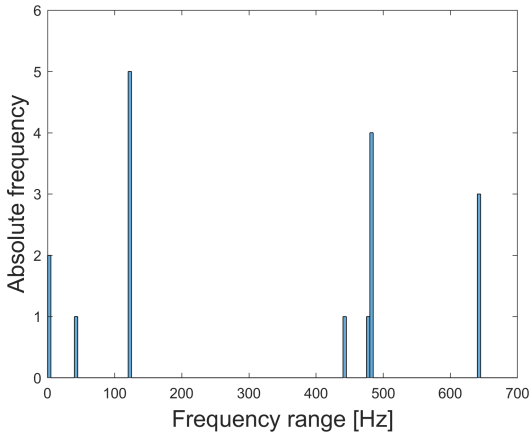


Figure 8. Frequency range of the 17 most relevant features. The frequencies of all relevant features are ≤ 640 Hz.

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 minimal possible time shift is ± 0.1 ms as the lowest sampling rate over all sensors is 10 kHz.

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

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

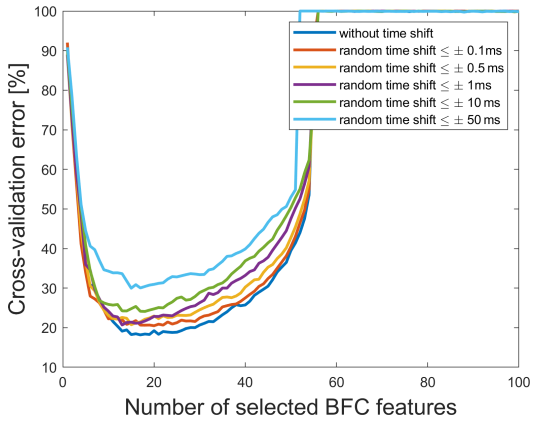


Figure 9. Cross-validation errors vs. the number of selected BFC features for different random simulated synchronization errors using RFESVM as selector.

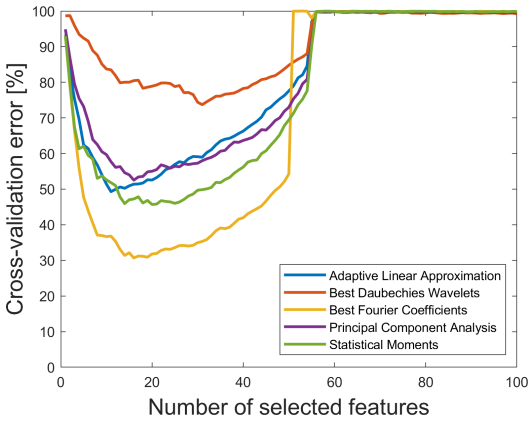


Figure 11. Cross-validation error vs. number of selected features for a maximum time shift of ±50 ms and RFESVM as selector. For a better visibility, only the results with RFESVM as selector are shown.

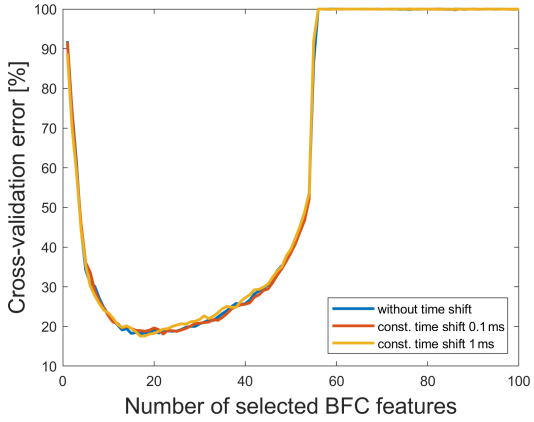


Figure 10. Cross-validation errors vs. the number of selected BFC features for constant shifted time windows with RFESVM as selector.

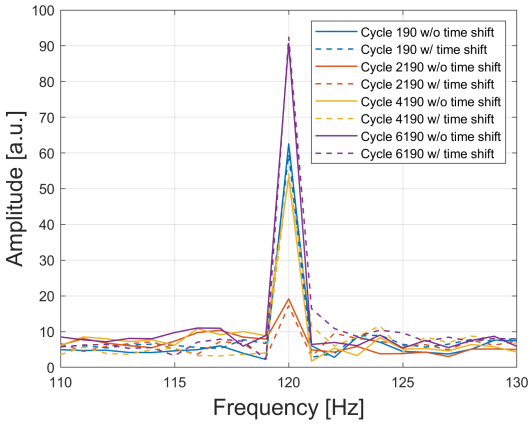


Figure 12. Best feature according to RFESVM (120 Hz of the active current) for the 2 kHz raw data set and the data set with random time shift of maximum 50 ms for three different cycles.

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.

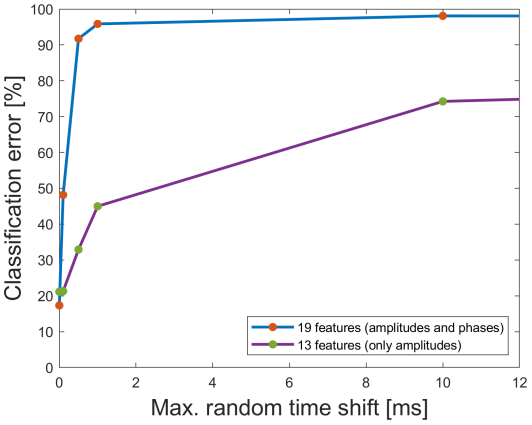


Figure 13. Classification error for one fold of the 10-fold cross validation using the raw data set for the model training and applying this model to data sets with different maximum random time shifts. Red dots represent models based on both amplitude and phase features, while green dots represent models 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

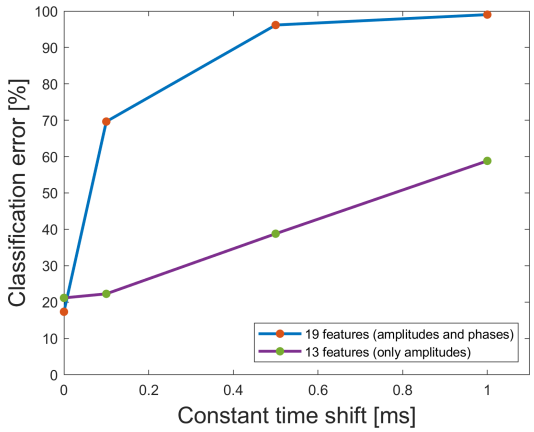


Figure 14. Classification error for one fold of the 10-fold cross validation using the 2 kHz raw data set for the model training and constant time-shifted data sets for the application of the trained model. Red dots represent models based on both amplitude and phase features, while green dots represent 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.

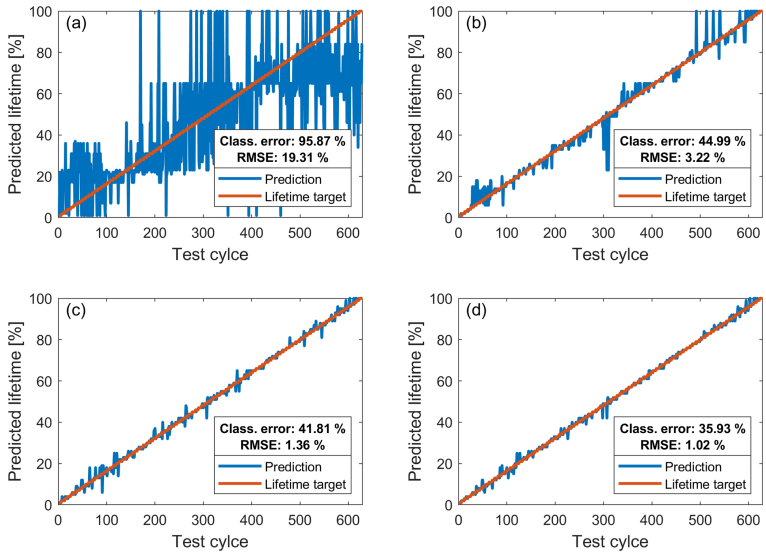


Figure 15. Predictions (blue) of the used EMC lifetime (steps of 1 %) for one fold of the 10-fold cross validation for the data set with time shifts of up to 1 ms and the assumed used lifetime target from 1 % to 100 % (red). **(a)** Model trained with raw data only using both amplitude and phase features. **(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.

3.5 Paper 3 – Uncertainty-aware automated machine learning toolbox

Quantitative measurements are carried out in industry every day. As no measurement result is exact, it is only complete with an accompanied quantitative statement of its associated uncertainty. Knowing the measurement uncertainty is crucial for assessing the reliability and comparability, and determining the quality of measurement results as well as evaluating the decisions based on these results. When decisions are based on machine learning inference, i.e., the process of applying a trained ML model to measurement data for obtaining decisions or predictions, assessing the reliability of the ML result, which is affected by the quality of the input measurement data, is essential. In addition to uncertainty in time presented in Section 3.4, uncertainties also occur in the measurement values. The AMLT, in its original version [38], does not consider any uncertainty in measurement values.

Paper 3 [222] presents an extended version of the AMLT, the so-called uncertainty-aware automated machine learning toolbox (UA-AMLT), for classification problems to overcome the limitation of neglected uncertainty in measurement values. This paper is based on the AMLT for classification problems in its original version, as presented in Figure 2.5a. Each ML method in the AMLT is equipped with uncertainty propagation for uncorrelated input quantities in line with the GUM and its supplements, Supplement 1 (GUM-S1) and Supplement 2 (GUM-S2).

Performing feature extraction, i.e., the mapping $\mathbf{D} \mapsto \mathbf{F_E}$ for every FE algorithm in the AMLT (cf. Section 2.2.1), not only the features are calculated but also the corresponding uncertainties. This leads to a matrix $\mathbf{U_{F_E}}$ of the same size as $\mathbf{F_E}$ containing the corresponding uncertainty values. For example, the associated uncertainty value for the measurement value $\mathbf{f_E}_{ij}$ is $\mathbf{u_{F_E}}_{ij}$. For uncertainty propagation through the *Principal Component Analysis* (PCA) algorithm, an efficient Monte Carlo method (MCM) implementation [223] is used based on GUM-S1 as an analytical approach leads to numerical issues for big data applications. The sensitivity coefficients for the remaining four FE algorithms are calculated according to the analytical approach (cf. Equation (2.2)) presented in the GUM. For *Adaptive Linear Approximation* (ALA) and BFC, the approaches had already been previously published in [40, 224, 225]. An uncertainty-aware *Best Daubechies Wavelets* (BDW) algorithm was already proposed in [41, 42]. This algorithm needs an adaption to the Daubechies Daubechies-4 ($D4$) wavelet, as shown in [226]. The missing *Statistical Moments* (SM) algorithm, i.e., the

calculation of mean, standard deviation, skewness, and kurtosis, and the corresponding analytical uncertainty propagation according to the GUM are presented in detail in Paper 3.

Expressed mathematically, feature selection is a mapping $\mathbf{F_E} \mapsto \mathbf{F_S}$ (cf. Section 2.2.2) that ranks features and removes redundant, irrelevant, and noisy ones according to the chosen FS algorithm. Paper 3 suggests using modified versions of the three FS algorithms, called weighted FS algorithms, which take uncertainty values into account for the ranking and removing process. In addition, a mapping $\mathbf{U_{F_E}} \mapsto \mathbf{U_{F_S}}$ is performed, whereby the uncertainty matrix $\mathbf{U_{F_S}}$ has the same size as $\mathbf{F_S}$ and contains the corresponding uncertainty values. As the FS process only ranks features and removes redundant, irrelevant, and noisy features from the feature matrices (cf. Section 2.2.2)), the associated uncertainty values of the excluded features contained in the uncertainty matrix are also removed. This means the FS step only rearranges features and their associated uncertainty and removes the not selected ones.

The classification step uses *Linear Discriminant Analysis* (LDA) as dimensionality reduction followed by a Mahalanobis distance classifier. The calculation of the uncertainty propagation for the LDA algorithm is based on GUM-S2, the extension of the GUM to any number of output quantities. The Mahalanobis distance is made uncertainty-aware by consideration of the vertices of a hyperrectangle. A hyperrectangle in the l-dimensional space consists of $2l$ facets, 2^l vertices, and $l \cdot 2^{l-1}$ edges [227]. l denotes the optimum number of the most relevant features (cf. Section 2.2.2). For a worst-case classification, as proposed in Paper 3, only points with a maximum possible distance from a projected point are relevant. The maximum possible distance is given by the uncertainty values, and thus, the vertices can be calculated by an addition/subtraction of the uncertainty values to the projected value. These relevant points, i.e., the vertices of a hyperrectangle, are shown in Figure 3.3 for the one-, two- and three-dimensional case.

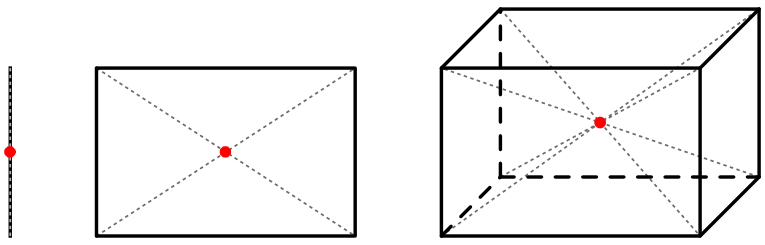
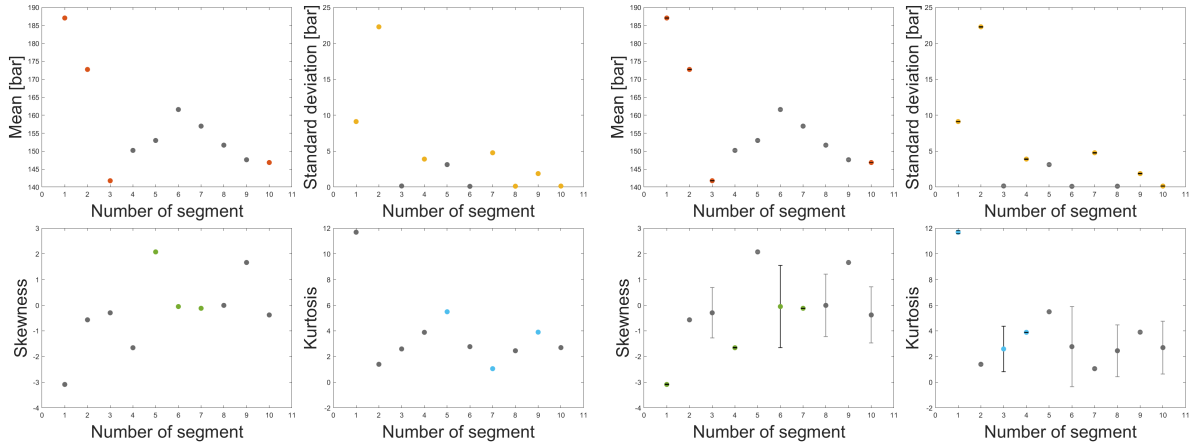


Figure 3.3: Hyperrectangle in case of one (line), two (rectangle), and three dimensions (cuboid) with an exemplary projected point (red) and the relevant distances (grey dashed line).

For every vertex of the hyperrectangle, the Mahalanobis distance to the central point of each class is calculated according to Equation (2.41). This leads to a class assignment for every vertex. The class assignments can be presented in a confusion matrix, as shown in Paper 3. For the worst-case classification, the minimum and maximum class value for every vertex is determined, leading to the worst-case class prediction for the vertices. This information is useful for a prediction plot, as shown in Paper 3.

For the demonstration of the UA-AMLT, the hydraulic system data set, which is presented in Section 2.3.1, is considered. An ML model is trained using data from the pressure sensor PS1 and the cooler efficiency as the target. Although the degradation of the cooler efficiency is a continuous process, the prediction of cooler efficiency is considered here as a classification problem where the target values are discrete steps (3 %, 20 %, and 100 % cooling efficiency), as explained in Section 2.3.1. Statistical Moments and weighted Pearson correlation are used as FE and FS algorithms, respectively. White noise with standard deviation $\sigma = 1\,\text{bar}$ is assumed as uncertainty contribution for the measured signals.

Figure 3.4 shows that the measurement uncertainty influences the selection of the most relevant features. In this example, the AMLT is applied to the same data as the



(a) 40 features extracted with Statistical Moments algorithm and the 17 most relevant ones (colored) according to Pearson correlation. Results obtained by using the AMLT.

(b) 40 features extracted with Statistical Moments algorithm and the 17 most relevant ones (colored) according to weighted Pearson correlation. Results obtained by using the UA-AMLT. Uncertainty values (represented as error bars) are analytically calculated with the UA-AMLT, as presented in Paper 3. Where no error bar is visible, the uncertainty value is in the hundredth bar pressure range.

Figure 3.4: 17 most relevant features (colored) and 23 less relevant features (greyed) determined using (a) the AMLT and (b) the UA-AMLT.

UA-AMLT, i.e., data from pressure sensor PS1 and cooling efficiency as target. In the AMLT, the shown features for one cycle are also calculated using Statistical Moments as FE algorithm. Their importance is determined by Pearson correlation in the AMLT and weighted Pearson correlation in the UA-AMLT. Considering, for example, the kurtosis features of the ten segments determined by the Statistical Moments algorithm, three features are chosen using the AMLT compared to using the UA-AMLT, whereby three others are chosen for the set of most relevant features. The skewness of the sixth segment is chosen by both toolboxes, the AMLT and the UA-AMLT, to be under the 17 most important features, although, in case of the UA-AMLT, this feature has the highest standard deviation of all skewness features. In the AMLT, this feature has the fifth-highest Pearson correlation coefficient considering all 40 features. Therefore, in the UA-AMLT, propagating the uncertainty through the FE algorithm still leads to a weighted Pearson correlation coefficient under the highest 17, i.e., this feature is still one of the 17 most important features despite the high uncertainty value.

This paper also shows with this example of FE and FS algorithm combination together with LDA and Mahalanobis distance as the classifier that measurement uncertainty for sensor data influences the model-based ML results. The benefit of uncertainty quantification for classification in ML is obtaining a more realistic class prediction. Moreover, weaknesses can be detected, e.g., noise susceptibility. For example, this information can be used to improve the ML model further.

DE GRUYTER OLDENBOURG tm–Technisches Messen 2023; 90(3): 141–153

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.

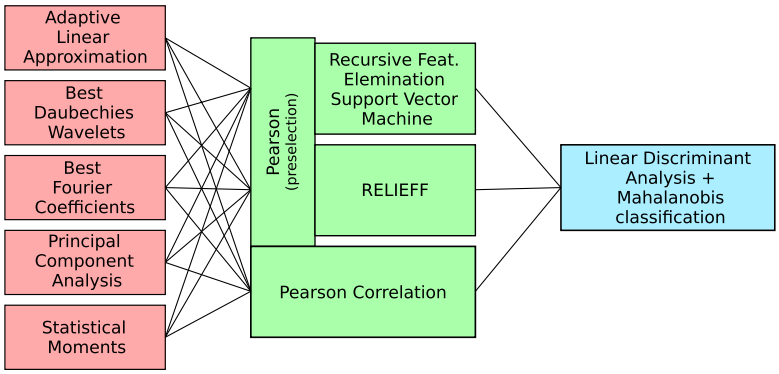


Figure 1: Scheme of the automated ML toolbox (AMLT) with feature extraction (red), feature selection (green) and classification (blue) (adapted from [2]).

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

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

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

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

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

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 $\mathbf{U_{F_E}}$ 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 $\mathbf{U_{F_E}}$ 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 $\mathbf{U_{F_E}}$ 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 $\mathbf{U}_{\mathbf{F_S}}$ 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 $\mathbf{U}_{\mathbf{F_S}}$ of the selected features be given by

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

where every feature in $\mathbf{F_S}$ the corresponding uncertainty value of $\mathbf{U}_{\mathbf{F_S}}$ is associated. The transpose matrix $\mathbf{U}_{\mathbf{F_S}}{}^\top$ 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 $\mathbf{L}$

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

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

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.

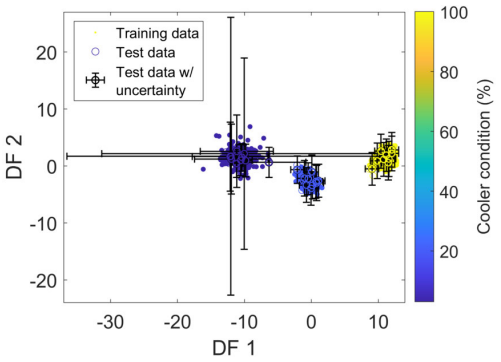


Figure 2: Uncertainty-aware LDA plot for training and test data. Uncertainty is presented as error bar only for every 5$^{\text{th}}$ test data point for better visibility.

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

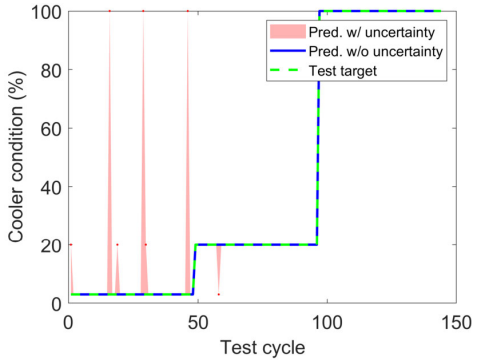


Figure 3: Prediction plot for test data with (red) and without (blue) consideration of uncertainty in contrast to the test target (green dashed).

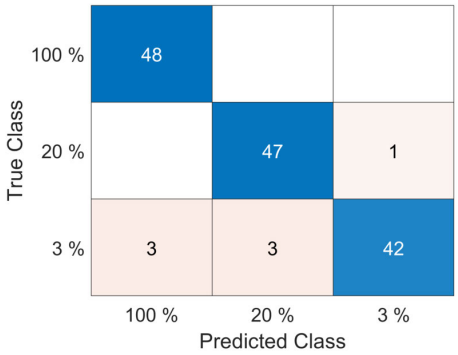


Figure 4: Confusion matrix for the cooler condition classification problem.

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.

3.6 Paper 4 – Influence of measurement uncertainty on machine learning results demonstrated for a smart gas sensor

Indoor air quality (IAQ) monitoring and thus detecting hazardous gases is critical to ensure human health and safety. IAQ, i.e., the concentrations of indoor air pollutants, can be predicted using a data-driven ML model. As gas concentrations are continuous quantities, they require regression which can be performed using the AMLT for regression problems (cf. Figure 2.5b). In ML, measurement uncertainty is usually not directly addressed for each prediction, but the average performance of the ML model is only estimated using CV [26, 27]. To obtain reliable and traceable results, measurement uncertainty has to be taken into account. For IAQ monitoring, it is essential that the threshold limit value (TLV) for the gas of interest is not reached even when measurement uncertainty is considered. Otherwise, the used sensor system is not useful for predicting the gas concentration of interest. Paper 4 [58] introduces the UA-AMLT for regression problems with which the influence of measurement uncertainty on ML results is investigated and demonstrated for the gas sensor data set presented in Section 2.3.3.

First, the AMLT for regression problems (cf. Figure 2.5b) has to be made uncertainty-aware to address measurement uncertainty directly. The AMLT for regression problems consists of five FE and one FS, as well as *Partial Least Squares Regression* (PLSR) for regression. As the five FE algorithms are the same as in the AMLT for classification problems, their uncertainty awareness is reached using the uncertainty propagation as explained in Paper 3 for each of the five algorithms. For FS, only Pearson correlation is included in the AMLT for regression problems, as described in Section 2.2. In the UA-AMLT for regression problems, weighted Pearson correlation is used as the uncertainty-aware FS algorithm. This algorithm ranks features according to their weighted Pearson correlation, in which the reciprocals of the squared associated uncertainty values of the features serve as weights. The FS step only rearranges features and their associated uncertainty and removes the not selected features and their associated uncertainty. Thus, only the uncertainty propagation for the PLSR algorithm needs to be developed to obtain the UA-AMLT for regression problems. An analytical approach to the uncertainty propagation for PLSR in line with GUM and its supplements is shown in detail in the presented paper. A benefit of this developed UA-AMLT is that the model must only

be trained once, and the associated uncertainty values can be propagated through the three toolbox steps in the model application case.

For the investigation of measurement uncertainty influence on ML results, which is carried out in Paper 4, the recorded initial calibration measurements performed with a metal oxide semiconductor (MOS) gas sensor SGP30 are used. These measurements are included in the gas sensor data set described in Section 2.3.3. Several RMSE values for assessment of an ML model and the influence of measurement uncertainty are introduced:

- test RMSE ($\mathrm{T-RMSE}$) results from applying the trained model to the test data,
- uncertainty RMSE ($\mathrm{U-RMSE}$), as the calculated difference of $\mathrm{T-RMSE}$ and $\mathrm{U-RMSE}$, denotes the RMSE resulting from propagating measurement uncertainty through the toolbox, and
- test plus uncertainty RMSE ($\mathrm{T+U-RMSE}$) is a measure of the quality of a model under the consideration of measurement uncertainty.

As target considered in Paper 4, formaldehyde, one of the most relevant carcinogenic gases indoors [172–174], is chosen on the one hand. On the other hand, the sum of the four volatile organic compounds (VOCs) as the indicator for IAQ, denoted as $\mathrm{VOC_{sum}}$, is chosen. As the results obtained for $\mathrm{VOC_{sum}}$ as target show the same trends and thus lead to the same conclusions as using formaldehyde as target, all further discussions are only based on the formaldehyde target.

As the AMLT for regression problems includes five FE algorithms, the FE algorithm, which leads together with FS and PLSR to the minimum CV error, i.e., the best FE for the given task, is determined. The used data set consists of 2,485 temperature cycles (TCs) during 497 unique gas mixtures (UGMs), as described in Section 2.3.3. To prevent the ML model from overfitting (cf. Section 2.2.4), the data set is split into training (70 %), validation (10 %), and test (20 %) data. Two different validation scenarios are considered: omitting complete UGMs (group-based CV) and randomly omitting individual TCs (random CV). Omitting complete UGMs implies that each of the UGMs is only present in one of the three data set splits at a time. In contrast to omitting complete UGMs, the automatically performed 10-fold stratified CV in the AMLT randomly omits individual TCs. In Paper 4, it is shown that there is a significant difference in the resulting CV error when using the two different validation scenarios. The random CV error is smaller when using PCA instead of ALA as the FE algorithm, whereas it is vise-versa for group-based CV error. Choosing the algorithm that leads

to the best group-based CV error is recommended, as the 10-fold stratified CV can not efficiently avoid overfitting in this application. For example, using formaldehyde as target, ALA is chosen as the best FE algorithm leading to a T − RMSE of 15.33 ppb for 20 PLSR components. This number of PLSR components is a good trade-off between model accuracy and computational cost, as shown in Paper 4. A slightly better result, i.e., a T − RMSE of 13.53 ppb, can be obtained using 100 PLSR components, but this needs more computational cost than using fewer PLSR components. Table 3.1 lists the different gases and the corresponding FE methods, which lead to the lowest group-based CV error resulting from a Monte Carlo simulation with 100 trials.

Table 3.1: Group-based CV error resulting from the best FE method for the different gases in ppb.

Substance	**FE method**	**Group-based CV error [ppb]**
Hydrogen	ALA	44.17 ± 6.11
Carbon monoxide	SM	91.36 ± 12.82
Acetone	ALA	16.55 ± 1.67
Ethanol	SM	35.81 ± 4.87
Formaldehyde	ALA	16.48 ± 2.21
Toluene	BDW	32.25 ± 4.05
VOC_{sum}	ALA	44.34 ± 6.86

Artificially generated additive noise is used to simulate measurement uncertainty at different signal-to-noise ratio (SNR) levels, as noise is one potential cause of measurement uncertainty (cf. Figure 2.1). Additive noise implies that the noise values are added to the measurement values. Two different noise models are considered in Paper 4: white Gaussian noise and white uniform noise. White noise means the values are statistically independent and uncorrelated, having zero mean and finite variance [228, 229]. The covariance matrix for white noise is given by $\sigma^2 I$, where σ^2 denotes the variance and I the identity matrix. For Gaussian noise, the amplitudes are modeled with a normal distribution [230], whereas for uniform noise, the uniform distribution [231] is used for modeling the amplitudes. The maximum theoretical SNR in decibel (dB) is determined according to [232], leading to 98 dB for the 16-bit analog-to-digital converter (ADC) of the SGP30 sensor. Therefore, SNRs between 0 dB and 98 dB with 5 dB increments are considered leading to a total of 21 SNR levels. As the results obtained for additive white uniform noise lead to the same conclusions as using additive white Gaussian noise, all further discussions are only based on the latter.

Two approaches are considered to investigate the influence of measurement uncertainty on ML results: model training with raw and noisy data, respectively. For the prediction of gas concentrations, the trained models are applied to noisy test data of varying SNR levels.

The first use case, i.e., model training with raw data, occurs when the precision of the sensor degrades over time, and no periodic recalibration is carried out. The model training with the raw data is carried out only once using the UA-AMLT for regression problems. For the application of noisy data, the associated uncertainty is propagated through the UA-AMLT for regression problems. No model retraining is necessary in case the uncertainty values change so that computational cost can be saved.

In Paper 4, it is shown that the model using the FE algorithm with the lowest group-based CV error, i.e., ALA, does not always perform best when applying test data of varying SNR levels. For SNR levels smaller than 50 dB, PCA performs best, meaning that this algorithm is able to compensate for noise in this decibel range. As demonstrated in Paper 4, noise as a source of measurement uncertainty directly influences the model performance, and thus, the FE method should be chosen with regard to the present SNR level. To make the best possible FE algorithm choice, training of five models for the five possible paths in the UA-AMLT as well as application of these trained models to noisy data of different SNR levels must be carried out, and depending on the lowest $\mathrm{T}+\mathrm{U}-\mathrm{RMSE}$, the best FE algorithm for the present SNR level is chosen.

For the 21 SNR levels studied for the first use case, it is also shown that while the influence of the RMSE resulting from the model ($\mathrm{T}-\mathrm{RMSE}$) is constant on the overall RMSE ($\mathrm{T}+\mathrm{U}-\mathrm{RMSE}$), as the training is carried out with the raw data, the influence of the measurement uncertainty ($\mathrm{U}-\mathrm{RMSE}$) decreases steadily with increasing SNR value. The constancy of the $\mathrm{T}-\mathrm{RMSE}$ across all SNR levels can be explained simply by the fact that the model was trained only once, and the uncertainty values are propagated through the toolbox.

Training a model with noisy data and applying this trained model to noisy data is the second use case. This use case can occur when using low-performance sensors or sensor systems that provide significant noisy data or where the ADCs add significant noise. The model is again trained using the UA-AMLT with ALA as the FE algorithm, but in this use case, the training is carried out with noisy data. Compared to the results of the first use case, it is shown that the $\mathrm{T}+\mathrm{U}-\mathrm{RMSE}$ is significantly smaller for the second use case, i.e., the training with noisy data. This leads to the assumption that the model is able to suppress noise if the training data already contains noise. Thus, it

is recommended to add generated noise of an SNR level higher than the noise of the raw data to the training data to achieve a more noise-resistant model.

Comparing both use cases, it is shown that the overall RMSE is significantly higher for the model trained with raw data than for the model trained with noisy data. This leads to the assumption that the ML model suppresses noise if the noise is already contained in the training data. Therefore, it is suggested to add artificially generated noise of smaller SNR levels than the noise of the raw data to the training data as the ML model gets more noise-resistant. For the 21 studied SNR levels, it can be seen that the $\mathrm{T}-\mathrm{RMSE}$ resulting from the model test is always larger than the $\mathrm{U}-\mathrm{RMSE}$ resulting from the noise within the data.

For both use cases, the analyses carried out in Paper 4 show two distinct possibilities where the overall measurement system can be improved to achieve better ML results with the UA-AMLT. On the one hand, the trained model and, on the other hand, the used sensor can be improved. In case of a high $\mathrm{U}-\mathrm{RMSE}$, optimizing the used sensor and the data acquisition electronics is recommended in Paper 4. In contrast, if the $\mathrm{U}-\mathrm{RMSE}$, i.e., the RMSE value resulting from propagating measurement uncertainty through the toolbox, tends towards zero, an improvement of the ML model should be considered as an improvement of the sensor system has no significant influence on the $\mathrm{T}+\mathrm{U}-\mathrm{RMSE}$. Model improvement, i.e., a reduction of $\mathrm{T}-\mathrm{RMSE}$, can be achieved using more PLSR components, as already shown, or by using deep learning based on

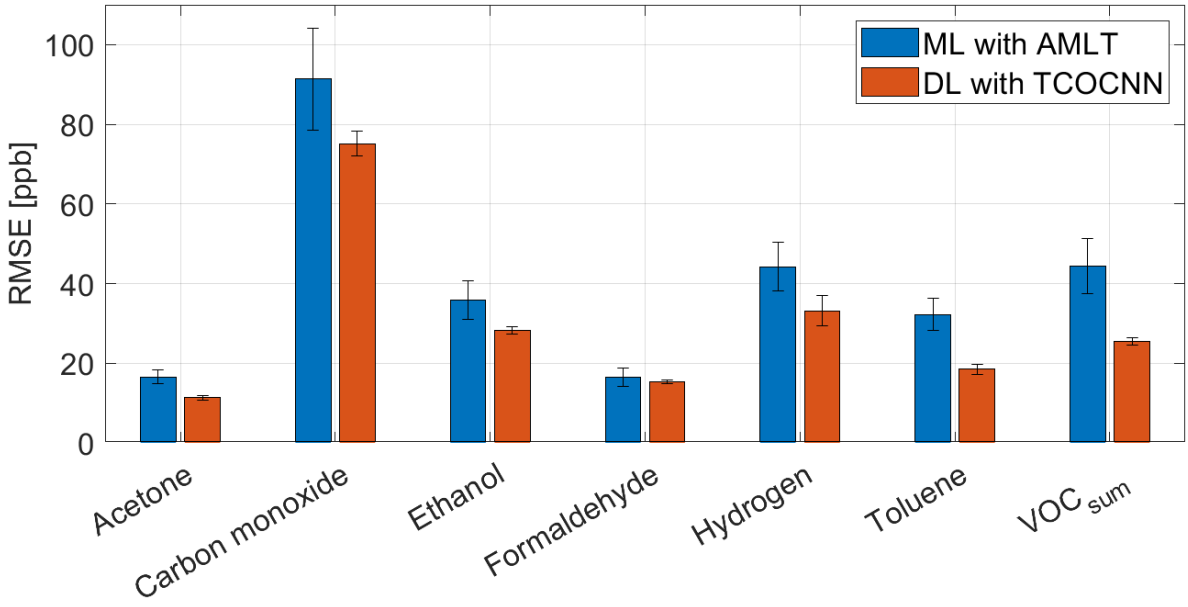


Figure 3.5: Comparison of the RMSE values obtained with deep learning [233] and the AMLT.

artificial neural networks (ANNs). Figure 3.5 compares the results achieved using the UA-AMLT to those obtained by using a 10-layer deep convolutional neural network (TCOCNN), as developed in [233]. Although DL leads to better prediction results for all gases, ML has the benefit of tracing a prediction made by an ML back, i.e., the interpretation of a model, is easier than for a DL model [25].

J. Sens. Sens. Syst., 12, 45–60, 2023
https://doi.org/10.5194/jsss-12-45-2023

JSSS Journal of Sensors and Sensor Systems Open Access

Influence of measurement uncertainty on machine learning results demonstrated for a smart gas sensor

Tanja Dorst[1,2], **Tizian Schneider**[1,2], **Sascha Eichstädt**[3], **and Andreas Schütze**[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]Fachbereich 9.4, Physikalisch-Technische Bundesanstalt, Braunschweig and Berlin, Germany

Correspondence: Tanja Dorst (t.dorst@lmt.uni-saarland.de)

Received: 1 August 2022 – Revised: 18 November 2022 – Accepted: 9 January 2023 – Published: 27 January 2023

Abstract. Humans spend most of their lives indoors, so indoor air quality (IAQ) plays a key role in human health. Thus, human health is seriously threatened by indoor air pollution, which leads to 3.8×10^6 deaths annually, according to the World Health Organization (WHO). With the ongoing improvement in life quality, IAQ monitoring has become an important concern for researchers. However, in machine learning (ML), measurement uncertainty, which is critical in hazardous gas detection, is usually only estimated using cross-validation and is not directly addressed, and this will be the main focus of this paper. Gas concentration can be determined by using gas sensors in temperature-cycled operation (TCO) and ML on the measured logarithmic resistance of the sensor. This contribution focuses on formaldehyde as one of the most relevant carcinogenic gases indoors and on the sum of volatile organic compounds (VOCs), i.e., acetone, ethanol, formaldehyde, and toluene, measured in the data set as an indicator for IAQ. As gas concentrations are continuous quantities, regression must be used. Thus, a previously published uncertainty-aware automated ML toolbox (UA-AMLT) for classification is extended for regression by introducing an uncertainty-aware partial least squares regression (PLSR) algorithm. The uncertainty propagation of the UA-AMLT is based on the principles described in the *Guide to the Expression of Uncertainty in Measurement* (GUM) and its supplements. Two different use cases are considered for investigating the influence on ML results in this contribution, namely model training with raw data and with data that are manipulated by adding artificially generated white Gaussian or uniform noise to simulate increased data uncertainty, respectively. One of the benefits of this approach is to obtain a better understanding of where the overall system should be improved. This can be achieved by either improving the trained ML model or using a sensor with higher precision. Finally, an increase in robustness against random noise by training a model with noisy data is demonstrated.

1 Introduction

1.1 Indoor air quality and VOCs

As humans spend most of their lives indoors, the most significant environment for them is the indoor environment (Brasche and Bischof, 2005). For this reason, indoor air quality (IAQ) is of special importance as it plays a leading role with regard to the performance, well-being, and health of humans (Sundell, 2004; Asikainen et al., 2016). Volatile organic compounds (VOCs) are one of the main contributors to poor air quality, especially in indoor air, and can lead to serious health problems, e.g., leukemia, cancers, or tumors (Jones, 1999; Tsai, 2019). Nowadays, IAQ monitoring is mostly based on measurements of carbon dioxide (CO_2) emitted by humans as the primary indicator for poor indoor air, as CO_2 concentration is directly related to VOCs caused by human presence (Von Pettenkofer, 1858). However, this neglects the fact that not only humans emit VOCs but also their activities such as household cleaning, cooking, and smoking, as well as, for example, furniture, carpets, and even the building itself due to the building materials used (Spaul, 1994). To measure almost all types of VOCs in in-

Published by Copernicus Publications on behalf of the AMA Association for Sensor Technology.

door air, metal oxide semiconductor (MOS) gas sensors are widely used as they are low-cost, robust, and highly sensitive. To improve the limited selectivity of these sensors and enable the discrimination of specific pollutants, MOS gas sensors can be operated in dynamic modes, especially by using a temperature-cycled operation (TCO; Eicker, 1977; Lee and Reedy, 1999; Baur et al., 2015; Schütze and Sauerwald, 2020a; Baur et al., 2021). A TCO, especially in combination with modern microstructured gas sensors, yields extensive and rich response patterns that need to be interpreted using machine learning (ML) to extract the relevant information (Schütze and Sauerwald, 2020a).

For the data set used in this contribution, sensor responses of an SPG30 sensor (Sensirion AG, Stäfa, Switzerland) with four gas-sensitive layers in TCO were recorded (Rüffer et al., 2018). This contribution focuses on formaldehyde as an example of a highly relevant toxic gas and on the sum concentration of all VOCs (VOC_{sum}) in parts per billion (ppb) in the used data set, i.e., the sum of the concentrations of acetone, ethanol, formaldehyde, and toluene. VOC_{sum} should not be confused with the widely used total VOC (TVOC) value, as this is based on analytical measurements and takes into account only VOCs with medium volatility (Schütze and Sauerwald, 2020b). Formaldehyde (CH_2O) is one of the most toxic and carcinogenic gases in indoor air (Hauptmann et al., 2004; Zhang, 2018; NTP, 2021) and is released from a variety of sources. The most significant ones are pressed wood products, e.g., particle board and plywood paneling. The World Health Organization (WHO) set the guideline threshold for a 30 min average concentration to 0.1 mg m^{-3}, which corresponds to approximately 80.1 ppb for 760 mmHg and 20 °C (World Health Organization, 2010).

1.2 Automated ML toolbox

In recent years, an automated machine learning toolbox (AMLT) was developed and applied to different classification tasks (Schneider et al., 2017, 2018; Dorst et al., 2021). Its extension to an uncertainty-aware AMLT (UA-AMLT) for classification was presented in Dorst et al. (2022). As gas concentrations are continuous quantities, regression must be used, which is a supervised ML technique. In this contribution, the AMLT is therefore extended to be applicable for regression tasks and, furthermore, the corresponding uncertainty for the ML result is considered. The uncertainty propagation is based on the *Guide to the Expression of Uncertainty in Measurement* (GUM; BIPM et al., 2008a) and its Supplement 1 (BIPM et al., 2008b) and Supplement 2 (BIPM et al., 2011). These three documents establish general rules for evaluating and expressing measurement uncertainty. These rules and principles are applied in this contribution for estimating the uncertainty of an ML model prediction, thus extending the GUM approach to smart sensors.

To investigate the influence of measurement uncertainty on machine learning (ML) results, sensor raw data are manipulated by simulated additive white Gaussian noise. With these manipulated data sets, different ML models are determined based on feature extraction, feature selection followed by regression, and the influence of the Gaussian noise, which simulates increased sensor uncertainty in the ML results, is investigated. Gaussian (normally distributed) noise is a very good assumption for any process for which the central limit theorem holds. In addition, the influence of additive white uniform noise as a further noise model is investigated.

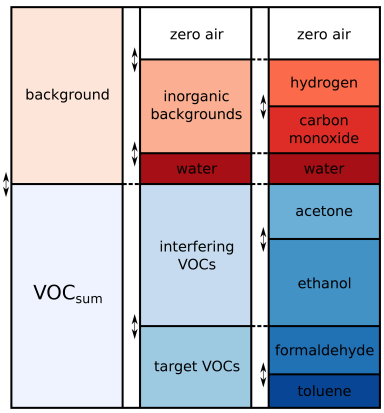


Figure 1. Gas composition for calibration consisting of random mixtures of VOCs (blue) and background gases (red; adapted from Baur et al., 2021).

2 Materials and methods

2.1 Data set

A data set published in Baur et al. (2021) is used to investigate the influence of measurement uncertainty on ML results. It consists of different calibration and field test measurements of gas mixtures with the MOS gas sensor SGP30 (Sensirion AG, 2020). The gas mixtures are composed of random mixtures of seven different gases that are relevant for indoor air quality. Various VOCs, i.e., acetone, ethanol, formaldehyde, and toluene, are used together with water vapor and inorganic background gases, i.e., hydrogen and carbon monoxide, as shown in Fig. 1. The gas concentrations are mixed using Latin hypercube sampling (LHS; McKay et al., 1979) to obtain unique gas mixtures (UGMs). In this contribution, only data from the initial calibration are used. The concentration ranges for all gases during the initial calibration are shown in Table 1.

The SGP30 sensor, with its four different gas-sensitive layers, is used in TCO to improve its selectivity, sensitivity, and stability (Schultealbert et al., 2018). As shown in Fig. 2, the temperature cycle consists of 10 steps at 400 °C, with a duration of 5 s each, followed by different low-temperature steps,

Table 1. Concentration ranges for all gases during the initial calibration phase (Amann et al., 2021b).

Substance	Minimum	Maximum
Humidity	25 % RH	70 % RH
Hydrogen	400 ppb	2000 ppb
Carbon monoxide	150 ppb	2000 ppb
Acetone	14 ppb	300 ppb
Ethanol	4 ppb	300 ppb
Formaldehyde	1 ppb	400 ppb
Toluene	4 ppb	300 ppb
VOC_{sum}	300 ppb	1200 ppb

RH is the relative humidity.

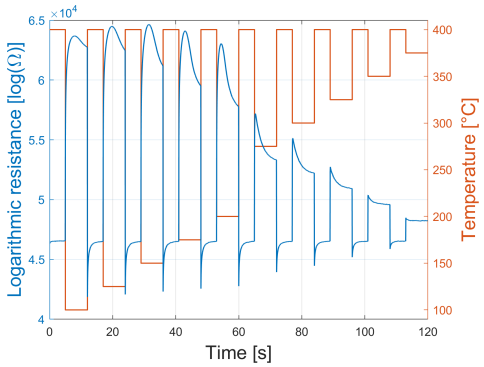


Figure 2. Logarithmic conductance of one sensor element (blue) and the temperature-cycled operation of the SGP30 (red).

with a duration of 7 s each. One single temperature cycle thus lasts 120 s and, due to the sampling rate of 20 Hz, consists of 2.400 measurement values for each gas-sensitive layer. The sensor output represents the logarithmic resistance shown for one cycle and one gas-sensitive layer in Fig. 2.

During the initial calibration phase, the SGP30 sensor is exposed to 500 UGMs for 10 temperature cycles (TCs) each. Due to the limited time response of the gas mixing apparatus (GMA) and synchronization problems between sensor and GMA, four TCs at the beginning and the last TC for each UGM are omitted so that only five TCs per UGM are evaluated. Furthermore, the first three UGMs are also not considered due to run-in effects. Thus, the data set comprises 2485 relevant cycles of 497 UGMs with stable gas concentrations from the initial calibration.

2.2 Uncertainty-aware automated machine learning toolbox

In general, regression is used for predicting a continuous quantity, whereas classification is used for predicting a discrete class label. As a basis for this publication, the AMLT

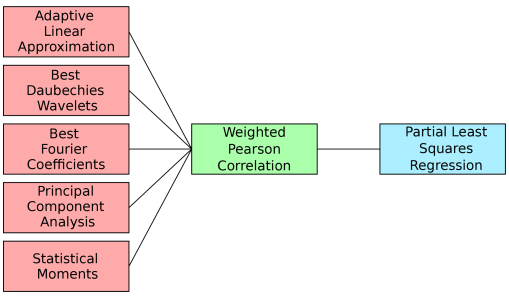


Figure 3. Feature extraction (red), feature selection (green), and regression (blue) algorithms of the uncertainty-aware AMLT for regression tasks.

for classification tasks (Schneider et al., 2017, 2018; Dorst et al., 2021) and its extended uncertainty-aware version (Dorst et al., 2022) are modified to also solve regression tasks. With the AMLT, feature extraction (FE) and feature selection (FS), as well as classification/regression and evaluation, are performed without expert knowledge and without a detailed physical model of the process to minimize model generation costs. Model training, in addition to application, can be carried out with the (uncertainty-aware) AMLT. Partial least squares regression (PLSR) as the de facto standard for quantification in the field of gas sensors (Wold et al., 2001; Gutierrez-Osuna, 2002) is used for regression tasks in the AMLT. Another well-known regression algorithm is principal component regression (PCR), which first performs the principal component analysis (PCA) as an unsupervised technique to obtain the principal components (PCs) and then uses these PCs to build the regression model. As a two-step model-building algorithm, the PCR makes interpreting the ML results harder in contrast to PLSR, which only has one step (Ergon, 2014). Using PCA leads to a relevant drawback of the PCR algorithm, as performing an unsupervised technique does not guarantee that the selected principal components for the regression model building are associated with the target. An advantage of PLSR is that it often has fewer components than PCR to achieve the same prediction level (De Jong, 1993a).

As shown in Fig. 3, five complementary FE algorithms are used within the AMLT, together with Pearson correlation for FS and PLSR, as the regression algorithm.

Adaptive linear approximation splits cycles into approximately linear segments, and for each segment, the mean value and slope are extracted as features from the time domain (Olszewski et al., 2001). The best Daubechies wavelets algorithm performs a wavelet transform using a Daubechies D4 wavelet (Daubechies, 1992) to extract 10 % of the wavelet coefficients with the highest average absolute value over all cycles as features from the time frequency domain. The best Fourier coefficients algorithm per-

forms a Fourier transform, and 10 % of the amplitudes with the highest average absolute value over all cycles and their corresponding phases are extracted from frequency domain (Mörchen, 2003). Using principal component analysis, projections on the principal components are determined (Pearson, 1901; Jackson, 1991) and used as features from the time domain. Moreover, the statistical distribution of the measurement values also includes information in the time domain (Martin and Honarvar, 1995). Thus, the cycles are split into 10 approximately equally sized segments, and the four statistical moments (mean, standard deviation, skewness, and kurtosis) are extracted for each segment as features. These five FE algorithms and the Pearson correlation as FS lead to five different algorithm combinations, each benchmarked to choose the best one for the respective application. The best combination is determined by the smallest cross-validated root mean square error (RMSE), which is a measure for the differences between the predicted $\boldsymbol{y}_{\mathbf{pred}} \in \mathbb{R}^m$ and the observed target values $\boldsymbol{y}$ of the same dimension, i.e.,

$$\mathrm{RMSE}(\boldsymbol{y}_{\mathbf{pred}}, \boldsymbol{y}) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_{\mathrm{pred}_i} - y_i)^2}\,. \tag{1}$$

The cross-validation (CV) scenario used is explained in Sect. 2.2. In general, different metrics can be used to describe the performance of a regression model; however, RMSE is one of the best interpretable error measures as it has the same unit as the prediction of the model and is also comparable to the (standard) measurement uncertainty used in describing data quality in measurement.

To use the UA-AMLT, a data matrix $\mathbf{D} \in \mathbb{R}^{m\times n}$ for each sensor (or sensor layer) must be given, where m denotes the number of cycles of length n. In case of non-cyclic sensor data, data must be windowed to obtain the correct $m \times n$ format. Furthermore, there must be knowledge about the uncertainty matrix $\mathbf{U} \in \mathbb{R}^{m\times n}$, which assigns an uncertainty value u_{ij} to a measurement value $d_{ij}\ \forall\ i, j$. This means that correlation of errors at different time instants is neglected.

Uncertainty-aware feature extraction and selection

To perform FE, which mathematically describes the mapping $\mathbf{D} \longmapsto \mathbf{F_E}$, five complementary methods are used. In this step, one feature matrix $\mathbf{F_E} \in \mathbb{R}^{m\times k}$, $k \leq n$ is calculated for each of the FE methods. The uncertainty calculation is performed according to Dorst et al. (2022), so that, for every feature matrix $\mathbf{F_E}$, an uncertainty matrix $\mathbf{U}_{\mathbf{F_E}}$ of the same dimension is calculated.

In the uncertainty-aware FS step, features are ranked according to their weighted Pearson correlation to the target value, i.e., in this contribution to the gas concentration. In weighted Pearson correlation, the reciprocals of the squared uncertainty values of the features are used as weights (Dorst et al., 2022). After ranking the features, a 10-fold stratified CV (Kohavi, 1995) is carried out for every possible number of features, and the minimum CV error is determined based on the optimal number of features $l \in \mathbb{N}$ found. From a mathematical point of view, FS is a mapping $\mathbf{F_E} \longmapsto \mathbf{F_S}$, with $\mathbf{F_S} \in \mathbb{R}^{m\times l}$, $l < k$ containing only the optimal number of the most relevant features according to weighted Pearson correlation. The corresponding uncertainty matrix is $\mathbf{U}_{\mathbf{F_S}} \in \mathbb{R}^{m\times l}$.

2.3 Partial least squares regression

Let a predictor matrix $\mathbf{X} \in \mathbb{R}^{m\times l}$ and a responses matrix $\mathbf{Y} \in \mathbb{R}^{m\times s}$ be given. The basic algorithm for computing a PLSR of $\mathbf{Y}$ on $\mathbf{X}$ using n_{comp} PLSR components is developed in Wold et al. (1984). Performing PLSR means iteratively solving the following decompositions, such that the covariance between $\mathbf{X}$ and $\mathbf{Y}$ is maximized as follows:

$$\mathbf{X} = \mathbf{X_S} \cdot \mathbf{X_L}^\top + \mathbf{X}_{\mathrm{res}} \tag{2}$$

$$\mathbf{Y} = \mathbf{Y_S} \cdot \mathbf{Y_L}^\top + \mathbf{Y}_{\mathrm{res}}, \tag{3}$$

where $\mathbf{X_L} \in \mathbb{R}^{l\times n_{\mathrm{comp}}}$ and $\mathbf{Y_L} \in \mathbb{R}^{s\times n_{\mathrm{comp}}}$ denote the orthogonal loading matrices. $\mathbf{X_S} \in \mathbb{R}^{m\times n_{\mathrm{comp}}}$ and $\mathbf{Y_S} \in \mathbb{R}^{m\times n_{\mathrm{comp}}}$ are the predictor and response scores, respectively. The matrices $\mathbf{X}_{\mathrm{res}}$ and $\mathbf{Y}_{\mathrm{res}}$ are the residual terms for predictor and response, respectively, and are used as a start for the next iteration step.

In MATLAB®, the partial least squares regression (PLSR) is calculated using the SIMPLS (statistically inspired modification of the partial least squares) algorithm (De Jong, 1993b). The advantage of SIMPLS is that the regression coefficients are determined directly without inverse matrices or singular value decomposition. Assume that $\hat{\mathbf{X}} \in \mathbb{R}^{m\times(l+1)}$ denotes a matrix in which a vector of ones is prepended to $\mathbf{X}$ to compute coefficient estimates for a model with constant terms. With $\mathbb{1} \in \mathbb{R}^m$ denoting a vector containing only ones, it holds for the augmented matrix that $\hat{\mathbf{X}} = (\mathbb{1} \,|\, \mathbf{X}) \in \mathbb{R}^{m\times(l+1)}$. The SIMPLS algorithm involves the calculation of a weighted matrix $\mathbf{W} \in \mathbb{R}^{(l+1)\times n_{\mathrm{comp}}}$. For the SIMPLS algorithm, the following holds:

$$\mathbf{X_S} = \hat{\mathbf{X}} \cdot \mathbf{W} \quad \text{and} \tag{4}$$

$$\mathbf{Y} = \mathbf{X_S} \cdot \mathbf{Y_L}^\top. \tag{5}$$

Combining Eqs. (4) and (5) leads to the following:

$$\mathbf{Y} = \hat{\mathbf{X}} \cdot \mathbf{W} \cdot \mathbf{Y_L}^\top \tag{6}$$

$$= \hat{\mathbf{X}} \cdot \mathbf{B}, \tag{7}$$

where $\mathbf{B} \in \mathbb{R}^{(l+1)\times s}$ denotes the matrix containing intercept terms in the first row and PLSR coefficient estimates in the others (De Jong, 1993b).

Uncertainty-aware partial least squares regression

In this contribution, the target values $\boldsymbol{y} \in \mathbb{R}^m$ are only represented by one vector, which leads to the matrix $\mathbf{B}$ (see

Sect. 2.3) being also only a vector $\boldsymbol{\beta} \in \mathbb{R}^{l+1}$. The matrix of selected features is given by $\mathbf{F_S} \in \mathbb{R}^{m \times l}$. $\hat{\mathbf{F}}_\mathbf{S} = (\mathbb{1} \,|\, \mathbf{F_S}) \in \mathbb{R}^{m \times (l+1)}$ denotes the matrix where one column of ones at the beginning of $\mathbf{F_S}$ was added. For PLSR, the following holds:

$$\boldsymbol{y}_\mathbf{pred} = \hat{\mathbf{F}}_\mathbf{S} \cdot \boldsymbol{\beta}, \tag{8}$$

with $\boldsymbol{y}_\mathbf{pred} \in \mathbb{R}^m$ representing the predicted target values. The basis of the uncertainty values calculation for the prediction $\boldsymbol{y}_\mathbf{pred}$ are formulas given in Sect. 6.2 ("Propagation of uncertainty for explicit multivariate measurement models") found in Supplement 2 of GUM (GUMS2; BIPM et al., 2011). This section of GUMS2 shows the covariance matrix calculation associated with an estimate of a multidimensional output quantity with the help of a sensitivity matrix using matrix–vector notation. This approach can be transferred to the propagation of uncertainty for PLSR. The first step is the transposing of Eq. (8), which leads to the following:

$$\boldsymbol{y}_\mathbf{pred}^\top = \boldsymbol{\beta}^\top \cdot \hat{\mathbf{F}}_\mathbf{S}^\top. \tag{9}$$

To use Sect. 6.2 of GUMS2, $\hat{\mathbf{F}}_\mathbf{S}$ and $\boldsymbol{\beta}$ must be transformed into vector and matrix, respectively. For the columns of $\hat{\mathbf{F}}_\mathbf{S}^\top$, the following holds:

$$\hat{\mathbf{F}}_\mathbf{S}^\top = \begin{pmatrix} 1 & \dots & 1 \\ \boldsymbol{f}_{S_1}^\top & \boldsymbol{f}_{S_2}^\top & \dots & \boldsymbol{f}_{S_m}^\top \end{pmatrix}, \tag{10}$$

where $\boldsymbol{f}_{S_i}^\top \in \mathbb{R}^l, \forall i = 1, \dots, m$ denotes the selected features for the ith cycle. Thus, the matrix–vector representation is given by the following:

$$\tilde{\boldsymbol{F}}_S^\top = \begin{pmatrix} 1 \\ \boldsymbol{f}_{S_1}^\top \\ 1 \\ \boldsymbol{f}_{S_2}^\top \\ \vdots \\ 1 \\ \boldsymbol{f}_{S_l}^\top \end{pmatrix} \in \mathbb{R}^{(m \cdot (l+1)) \times 1}, \tag{11}$$

and $\tilde{\boldsymbol{\beta}}^\top \in \mathbb{R}^{m \times (m \cdot (l+1))}$, with

$$\tilde{\boldsymbol{\beta}}^\top = \begin{pmatrix} \boldsymbol{\beta}^\top & 0 \dots 0 & 0 \dots 0 & \dots & 0 \dots 0 \\ 0 \dots 0 & \boldsymbol{\beta}^\top & 0 \dots 0 & \dots & 0 \dots 0 \\ 0 \dots 0 & 0 \dots 0 & \boldsymbol{\beta}^\top & \dots & 0 \dots 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 \dots 0 & 0 \dots 0 & 0 \dots 0 & \dots & \boldsymbol{\beta}^\top \end{pmatrix}, \tag{12}$$

which leads to

$$\boldsymbol{y}_\mathbf{pred}^\top = \tilde{\boldsymbol{\beta}}^\top \cdot \tilde{\boldsymbol{F}}_S^\top. \tag{13}$$

To propagate the uncertainty in the PLSR, the uncertainty matrix of the selected features $\mathbf{U_{F_S}} \in \mathbb{R}^{m \times l}$ must be extended with a column associated with the first column of $\hat{\mathbf{F}}_\mathbf{S}$. Thus, it holds that $\hat{\mathbf{U}}_{\mathbf{F_S}} = (\mathbb{0} \,|\, \mathbf{U_{F_S}}) \in \mathbb{R}^{m \times (l+1)}$. The transpose matrix $\hat{\mathbf{U}}_{\mathbf{F_S}}^\top$ is transferred to the diagonal matrix $\tilde{\mathbf{U}}_{\mathbf{F_S}}^\top \in \mathbb{R}^{(m \cdot (l+1)) \times (m \cdot (l+1))}$, where the rows of $\hat{\mathbf{U}}_{\mathbf{F_S}}$ are in the diagonal. Using Sect. 6.2.1.3 of BIPM et al. (2011) leads to the following:

$$\tilde{\mathbf{U}} = \tilde{\boldsymbol{\beta}}^\top \cdot \left(\tilde{\mathbf{U}}_{\mathbf{F_S}}^\top\right)^2 \cdot \left(\tilde{\boldsymbol{\beta}}^\top\right)^\top \tag{14}$$

$$= \tilde{\boldsymbol{\beta}}^\top \cdot \left(\tilde{\mathbf{U}}_{\mathbf{F_S}}^\top\right)^2 \cdot \tilde{\boldsymbol{\beta}}, \tag{15}$$

with $\tilde{\mathbf{U}} \in \mathbb{R}^{m \times m}$. To obtain the diagonal elements of $\tilde{\mathbf{U}}$, Eq. (15) can be simplified and retransformed to the following:

$$\mathbf{U}_\mathrm{PLSR}^\top = \left(\boldsymbol{\beta}^\top \circ \boldsymbol{\beta}^\top\right) \cdot \left(\hat{\mathbf{U}}_{\mathbf{F_S}}^\top \circ \hat{\mathbf{U}}_{\mathbf{F_S}}^\top\right) \tag{16}$$

$$\Leftrightarrow \quad \mathbf{U}_\mathrm{PLSR} = \left(\hat{\mathbf{U}}_{\mathbf{F_S}} \circ \hat{\mathbf{U}}_{\mathbf{F_S}}\right) \cdot (\boldsymbol{\beta} \circ \boldsymbol{\beta}) \tag{17}$$

$$= \hat{\mathbf{U}}_{\mathbf{F_S}}^{\circ 2} \cdot \boldsymbol{\beta}^{\circ 2}, \tag{18}$$

where $\circ$ denotes the Hadamard (element-wise) product (Horn, 1990). The uncertainty values associated with $\boldsymbol{y}_\mathbf{pred}$ can be calculated by the following:

$$\boldsymbol{U}_{\boldsymbol{y}_\mathbf{pred}} = \left(\left|\hat{\mathbf{U}}_{\mathbf{F_S}}^{\circ 2} \cdot \boldsymbol{\beta}^{\circ 2}\right|\right)^{\circ 1/2} \in \mathbb{R}^{m \times 1}, \tag{19}$$

where $|.|$ denotes the element-wise absolute value and $(.)^{\circ 1/2}$ the Hadamard (element-wise) square root (Reams, 1999).

3 Investigation of the influence of measurement uncertainty on ML results

To evaluate the influence of measurement uncertainty on ML results, the logarithmic resistance raw data of each sensor layer are modified by artificially generated additive white Gaussian noise of different signal-to-noise ratios (SNRs). This means that the logarithmic amplifier of the sensor is responsible for the noise. In general, the SNR is defined as the ratio of signal power to background noise power. SNR > 0 dB indicates that there is more signal than background noise. The maximum theoretical SNR in decibel (dB) for an analog-to-digital converter (ADC) can be determined, according to Bennett (1948), with the following:

$$\mathrm{SNR}(N) = 20 \cdot \log_{10}\left(2^N \cdot \sqrt{\frac{3}{2}}\right) \quad [\mathrm{dB}] \tag{20}$$

$$\approx 6.02 \cdot N + 1.76 \quad [\mathrm{dB}], \tag{21}$$

where N is the resolution of an ADC in bits. Thus, the maximum theoretical SNR for the 16 bit ADC of the SGP30 is approx. 98 dB. For this reason, only SNRs from 0 to 98 dB are considered in this publication. Figure 4 shows an example of raw and modified sensor data with different SNR values. The

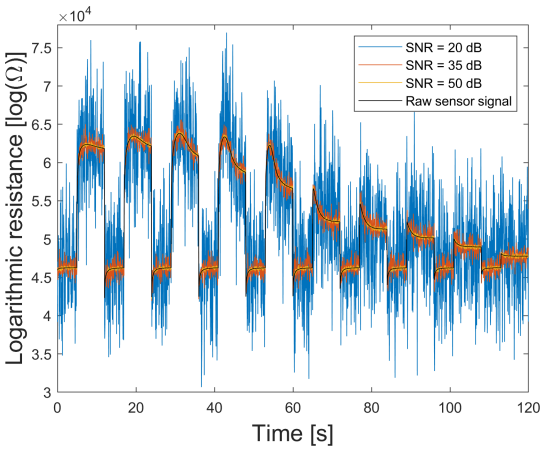


Figure 4. Raw (violet) and modified sensor signals with additive white Gaussian noise of different SNR values.

relation between the SNR and squared standard uncertainty σ^2 is given by the following:

$$\sigma^2 = 10^{\frac{\mathrm{SP}-\mathrm{SNR}}{10}}, \tag{22}$$

where the signal power (SP) is calculated by

$$\mathrm{SP} = 10 \cdot \log_{10}\left(\frac{\|\mathbf{A}\|_2^2}{m \cdot n}\right). \tag{23}$$

Here, $\mathbf{A} \in \mathbb{R}^{m \times n}$ denotes the data for one sensor. Thus, for example, 75 dB corresponds to $\sigma^2 = 91.71$, 80 dB to $\sigma^2 = 29.00$, and 98 dB to $\sigma^2 = 0.46$ for the first gas-sensitive layer of the SGP30. In practical applications, 98 dB is typically not reached because the measurement range of the ADC is larger than the range of actual measured values within the data set.

3.1 Application of AMLT

To investigate the influence of measurement uncertainty on machine learning results, the best FE algorithm must first be determined. To train, validate, and test a model, the data set is randomly split into 70 % training, 10 % validation, and 20 % test data by omitting complete UGMs in the training, validation, or test data set, respectively. This means that each of the 497 UGMs exists in either the training, validation, or test data but not in more than one at a time (see Fig. 5). Training the model is carried out by using the AMLT together with the training data and formaldehyde or $\mathrm{VOC_{sum}}$, respectively, as the target. The results obtained for $\mathrm{VOC_{sum}}$ as the target show the same trends and lead to the same conclusions as the results with formaldehyde as target and are therefore only shown in Sect. A2.

A 10-fold stratified CV is automatically performed in the AMLT to determine the best FE algorithm out of five

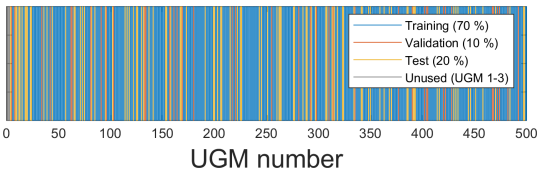


Figure 5. Randomized split of the UGMs into training, validation, and test data used in this contribution.

complementary FE methods. In contrast to the data split, which is carried out by omitting complete UGMs and used for performing group-based CV with validation data, the 10-fold stratified CV randomly omits individual TCs. The RMSE value resulting from the 10-fold CV is called the random CV error. To obtain quality information on the trained model, the differences between the predicted and the observed target values are measured using RMSE. The test RMSE (T − RMSE) results from applying the trained model to the test data. There can be a significant difference between a group-based CV error and T − RMSE, as the omitted UGMs are selected randomly. For each of the five algorithm combinations (see Fig. 3), a Monte Carlo different train, validation, and test data sets, is performed. The mean value and standard deviation are calculated for the three different errors resulting from using training, validation, and test data, with $n_{\mathrm{comp}} = 20$ in the PLSR algorithm. The reason for choosing $n_{\mathrm{comp}} = 20$ is given below. The results are shown in Fig. 6. Although principal component analysis (PCA) achieves the lowest random CV error mean value (14.7 ppb), with negligible variations for different splits and therefore seems to be the best FE algorithm, applying 10 % validation data will lead to a group-based CV error mean value of 24.7 ppb. This means that 10-fold stratified CV does not efficiently detect overfitting for this application as it does not omit complete UGMs and, thus, does not need to interpolate to different gas concentrations. A new, unpublished version of the AMLT already allows the user to define validation scenarios (random or group based). Here, adaptive linear approximation (ALA) as the second-best method with a random CV error mean value of 15.3 ppb is chosen for further investigations as there is no significant difference in the error mean values between omitting single TCOs (15.3 ppb; random validation) and complete UGMs (16.5 and 16.6 ppb; group-based validation). Thus, it is sufficient to evaluate the random CV error with 80 % training (including 10 % validation data) and 20 % test data split by omitting complete UGMs in the training or test data, respectively. Applying this trained model ($n_{\mathrm{comp}} = 20$; 80 % data used for model training) to the 20 % test data (see Fig. 5) leads to a shown in Fig. 7.

For $\mathrm{VOC_{sum}}$ as the target, the results are similar, and again, ALA is chosen as the best FE algorithm (random CV error mean value of 40.9 ppb) due to the overfitting of the trained model when using PCA (random CV error mean value of

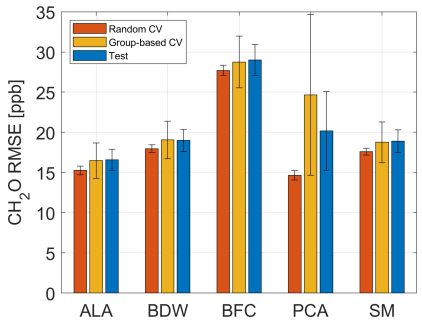


Figure 6. Random CV, group-based CV, and test RMSE of the five FE algorithms using Pearson as FS and PLSR with $n_{\text{comp}} = 20$ for 100 trials with different randomized UGM splits and the formaldehyde concentration as target. ALA is the adaptive linear approximation, BDW is the best Daubechies wavelets, BFC is the best Fourier coefficients, PCA is the principal component analysis, and SM is the statistical moments.

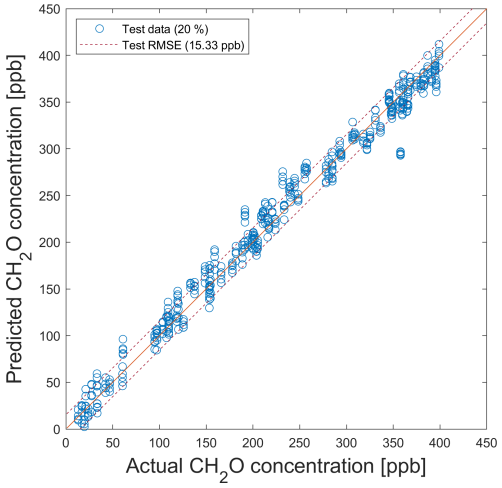


Figure 7. PLSR model for the quantification of formaldehyde for testing with test data from the data split shown in Fig. 5. Dashed lines indicate the RMSE of the test data (T – RMSE).

31.9 ppb; group-based CV error mean value of 61.3 ppb). Applying the model trained with the data split in Fig. 5 to the 20 % test data results in a T – RMSE of 46.1 ppb. The corresponding results are shown in Figs. A3 and A4.

To determine the optimal number of PLSR components, a Monte Carlo simulation (10 trials with different train and test data) was carried out, and the T – RMSE mean values of 10 trials, in addition to the corresponding standard deviations, were calculated. In Fig. 8, the T – RMSE value is plotted over the number of PLSR components for ALA as FE, Pearson as FS, and PLSR. For a small number of PLSR components, T – RMSE mean values have large standard deviations, for example, the standard deviation for $n_{\text{comp}} = 1$ is $\sigma = 5.5$ ppb. If the number of PLSR components is greater than 10, then the standard deviations are in the range from 1.15 to 1.22 ppb; thus, the obtained models are highly reproducible. The lowest T – RMSE mean value is achieved for a high number of PLSR components (here 13.5 ppb is achieved with $n_{\text{comp}} = 100$), but it is preferable to find a good trade-off between the accuracy and computational cost, as a lower number of PLSR components reduces the computational effort. Therefore, the optimal number of PLSR components is determined using the elbow method (Thorndike, 1953) to ensure a stable model, with a T – RMSE of 15.3 ppb. The elbow point, i.e., the point after which no further significant change occurs, is determined by using the ALA algorithm. ALA automatically determines four segments as being the best segmentation of the T – RMSE curve (see Fig. 8). Thus, the optimal number of PLSR components is $n_{\text{comp}} = 20$, as more components have no considerable influence on the T – RMSE, leading to higher computational cost and also increasing the risk of overfitting.

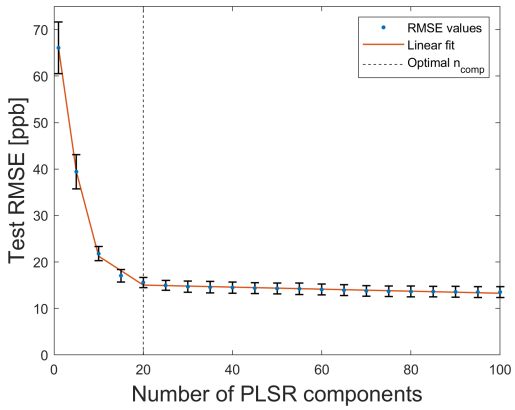


Figure 8. Elbow method applied to the T – RMSE curve for 10 trials. The optimal number of PLSR components is 20.

3.2 Influence of measurement uncertainty on ML results

In this contribution, two approaches for investigating the influence of the measurement uncertainty on machine learning results are considered, namely training a model with raw (see Sect. 3.2.1) and noisy (see Sect. 3.2.2) data, respectively. The trained models are used to predict data with varying noise levels between 0 and 98 dB in both use cases. Training a model with raw data means that the uncertainty associated with the raw data is propagated through the UA-AMLT, which saves on computational cost, as no retraining is necessary if the uncertainty changes. To validate the UA-AMLT, training with the noisy data of different SNRs is car-

ried out and compared to the results of the uncertainty propagation approach. The number of PLSR components ($n_{\text{comp}} = 20$) determined with the elbow method is considered for formaldehyde and VOC_{sum} as the target. For formaldehyde, the number of PLSR components leading to the minimum T − RMSE value, i.e., $n_{\text{comp}} = 100$, is also considered and compared to the results for

3.2.1 Model trained with raw data

The motivation for using raw data for training and noisy data for model application is the typical degradation of sensors over time (Jiang et al., 2006). To avoid a loss in sensor performance, periodical recalibration is typically required, which is often expensive and difficult or impossible to perform, as collecting sensors and sending them to the lab leads to the downtime of the IAQ monitoring system.

The test plus uncertainty RMSE (T + U − RMSE) is introduced as measure for the quality of the model considering the uncertainty values. This T + U − RMSE value is the sum of the two RMSE values obtained by the test of the model (T − RMSE) and by propagating the measurement uncertainty through the toolbox (U − RMSE), respectively. It is calculated according to the following:

$$\boldsymbol{d_u} = \boldsymbol{y_{\text{pred}}} + \boldsymbol{U}_{\boldsymbol{y_{\text{pred}}}} \tag{24}$$

$$\boldsymbol{d_l} = \boldsymbol{y_{\text{pred}}} - \boldsymbol{U}_{\boldsymbol{y_{\text{pred}}}} \tag{25}$$

$$\text{RMSE}_{\text{T+U}} = \begin{cases} \text{RMSE}(\boldsymbol{d_u}, \boldsymbol{y}) & \boldsymbol{y_{\text{pred}}} \geq \boldsymbol{y} \\ \text{RMSE}(\boldsymbol{d_l}, \boldsymbol{y}) & \text{otherwise} \end{cases}, \tag{26}$$

where $\boldsymbol{y} \in \mathbb{R}^m$ and $\boldsymbol{y_{\text{pred}}} \in \mathbb{R}^m$ denote the actual and the predicted target, respectively. $\mathbf{U}_{\boldsymbol{y_{\text{pred}}}}$ contains the uncertainty values associated with the predicted target. Furthermore, noisy data RMSE (ND-RMSE) is used, which indicates the quality of the model when applying it to another simulated data set (2000 cycles) with the added white Gaussian noise (noisy data) of different SNRs.

First, it is of interest if the selected FE algorithm still performs well when applying the model trained with raw data on noisy test data. ALA was chosen as the best FE algorithm when applying a model trained with raw data on raw test data, as shown in Sect. 3.1. Applying the model on noisy test data leads to the T + U − RMSE curves, as shown in Fig. 9. For SNR values greater than 65 dB, ALA achieves the smallest T + U − RMSE. In this range, the statistical moments (SM) also perform well, with the best Daubechies wavelets (BDW) achieving similar results for very high SNR ≤ 85 dB. The T + U − RMSE difference between ALA (best algorithm) and SM is only 1.9 ppb for 98 dB. Between 50 and 65 dB, the smallest T + U − RMSE is achieved using statistical moments. If SNR ≤ 50 dB, then PCA achieves the smallest T + U − RMSE. This means that PCA can compensate for noise in this range, but overfitting leads to higher error, as shown above for the raw data. This figure shows that the

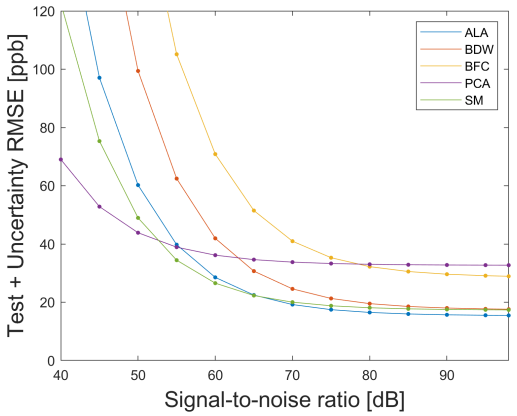


Figure 9. Test plus uncertainty RMSE (T + U − RMSE) curve for the five complementary FE algorithms, each in combination with the Pearson correlation for FS and PLSR.

measurement uncertainty has a direct influence on the performance of the ML algorithm, and thus, different FE methods should be chosen for different SNR values.

Figure 10a shows T − RMSE, T + U − RMSE, and ND-RMSE values for a model trained on raw data for SNR ≥ 40 dB (approx. maximum theoretical SNR for a 6 bit ADC, according to Eq. 21), using the data split shown in Fig. 5 with ALA as FE, the Pearson correlation for FS, and PLSR. The T − RMSE values in Fig. 10 are constant because the model was trained with one specific raw data split (see Fig. 5). For large SNR values, it can be assumed that the added white Gaussian noise is smaller than the SNR of the raw data and, therefore, has no significant influence, as is indeed observed. The T + U and ND errors show a similar increase with reduced SNR for both models with 20 and 100 PLSR components, with the ND error being slightly lower than the T + U error. This indicates that the model uncertainty estimated by propagating the error through the toolbox, i.e., the T + U error, overestimates the true model uncertainty slightly but still provides valuable insight into the sensitivity of the ML model to noisy data. To obtain an accurate model for predicting formaldehyde concentrations with $n_{\text{comp}} = 20$, the SNR of the data set should not fall below 70 dB, as, for this SNR, the T + U − RMSE is approx. 19.2 ppb, which is an acceptable uncertainty for determining the formaldehyde concentration with a threshold limit value (TLV) of 81 ppb. An SNR of 45 dB for $n_{\text{comp}} = 20$ and of 50 dB for $n_{\text{comp}} = 100$ results in T + U − RMSE and ND-RMSE values of approx. 80 ppb, i.e., similar to the TLV, which means that the sensor system would no longer be useful for estimating the formaldehyde concentration. For SNR < 80 dB, the model based on 20 PLSR components is more robust against noise than a model with 100 PLSR components, yielding lower RMSE values. In contrast, for SNR ≥ 80 dB, a higher number of

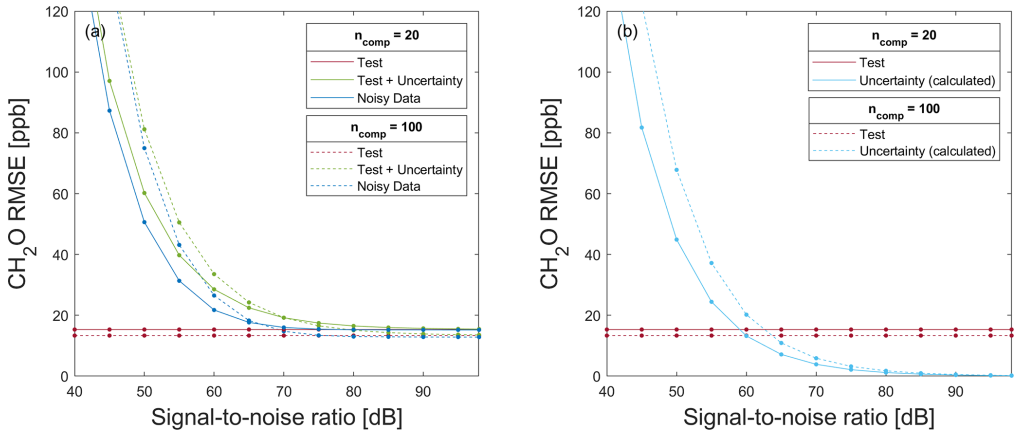


Figure 10. RMSE for testing a model trained with 80 % raw data for formaldehyde prediction on **(a)** 20 % test data without (red) and with associated uncertainty values (green), in addition to the application of the model on a noisy data set (blue) and **(b)** 20 % raw test data (red), and the calculated uncertainty RMSE (light blue) resulting from the difference in T + U − RMSE and T − RMSE for a different number of PLSR components.

PLSR components performs slightly better, i.e., an improvement in the RMSE values can be achieved with more PLSR components if the noise level in the data is very low. In the case of an SNR value of 98 dB, the T − RMSE value is 15.33 and 13.34 ppb for $n_{comp} = 20$ and $n_{comp} = 100$, respectively.

Figure 10b shows the T − RMSE and the U − RMSE. U − RMSE is calculated as the difference between T + U − RMSE and T − RMSE (see Fig. 10a). This figure shows that the influence of the trained model (expressed by T − RMSE) on T + U − RMSE is constant, while the influence of the measurement uncertainty (expressed by U − RMSE) decreases steadily with increasing SNR. For $n_{comp} = 20$ ($n_{comp} = 100$), U − RMSE is smaller than T − RMSE when SNR is greater than 60 dB (65 dB).

The results for the additive white uniform noise and formaldehyde as target are nearly the same as for the additive white Gaussian noise (see Fig. A8a). Similar results for VOC_{sum} as the target are shown in Fig. A5a.

To demonstrate the effect of the noise on test data, PLSR models trained with raw data ($n_{comp} = 20$) for the quantification of formaldehyde and VOC_{sum} are shown in Figs. A1 and A6 for the two different SNR values, respectively.

3.2.2 Model training with noisy data

The second use case occurs when using low-performance sensors or sensor systems that provide significant noisy data or where the electronics/ADCs add significant noise. For the investigation of the influence of measurement uncertainty on regression results, ALA as FE and Pearson correlation as FS are used together with PLSR. Formaldehyde as the target is discussed here, as VOC_{sum} leads to similar results, which are shown in Appendix A2. Only results for white Gaussian noise are shown here, as the results for white uniform noise are similar (see Appendix A3).

Figure 11a shows T − RMSE, T + U − RMSE, and ND-RMSE values for a model trained on noisy data for SNR ≥ 40 dB, using the data split shown in Fig. 5. Compared to Fig. 10a, the T + U − RMSE is significantly smaller for the model trained with noisy data, i.e., the ML model can suppress noise if it is contained in the training data. For example, for SNR = 50 dB and 20 PLSR components, the T + U − RMSE is 60.22 ppb when training with raw data, while it is only 34.3 ppb when training with noisy data. In general, a model can be made more noise resistant by adding additive white Gaussian noise to the training data. Comparing Figs. 10a and 11a, note that the regression results are similar for noisy and raw data for SNR ≥ 80 dB, thus indicating again that the noise level of the raw data is approx. 80 dB. The same holds for T + U − RMSE when SNR ≥ 80 dB. Of course, there is no need to train the model with noisy data with an added noise level lower than the noise of the original data. As already observed for the model trained with raw data, the RMSE values can be reduced by using more components, but here, this observation holds for all SNR levels, as the noise is contained in the training data, so there is no overfitting. This means that training a model with raw data once is sufficient, and no new model must be trained with noisy data. The associated measurement uncertainty values must only be used in the application of the model, which saves much computational cost. Figure 11b shows that, for T + U − RMSE values, the contribution resulting from the measurement uncertainty is always lower than the contribution from the test of the model. This means that the noise is already trained in

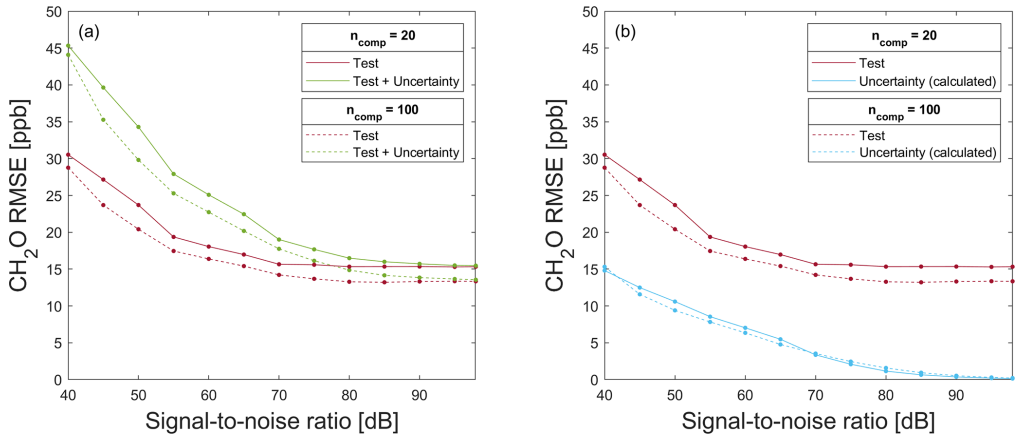


Figure 11. RMSE for testing a model trained with 80 % noisy data for formaldehyde prediction on **(a)** 20 % test data without (red) and with associated uncertainty values (green) and **(b)** 20 % raw test data (red) and the calculated uncertainty RMSE (light blue) resulting from the difference in T + U − RMSE and T − RMSE for a different number of PLSR components. Note the magnification compared to Fig. 10.

the model, and improving the used sensor would significantly improve the ML results.

For white uniform noise, similar results are shown in Fig. A8b.

In case of VOC_{sum} as the target, the results are similar, despite the fact that the RMSE values are higher than for formaldehyde, as shown in Fig. A5b. No significant difference between the RMSE values when training with raw and noisy data, respectively, is observed for SNR values higher than 70 dB.

To demonstrate the effect of noise on test data, PLSR models trained with noisy data ($n_{comp} = 20$) for the quantification of formaldehyde and VOC_{sum} are shown in Figs. A2 and A7, for two different SNR values, respectively.

4 Conclusion and outlook

In this contribution, the uncertainty-aware AMLT for classification tasks presented in Dorst et al. (2022) was first extended for solving regression problems. In accordance with the GUM, an analytical method for uncertainty propagation of PLSR was implemented. The code for this UA-AMLT for classification and regression tasks was published on GitHub (https://github.com/ZeMA-gGmbH/LMT-UA-ML-Toolbox, last access: 18 January 2023). For different SNR levels, the UA-AMLT automatically selects the best ML algorithm based on the overall test plus the uncertainty RMSE.

The influence of measurement uncertainty on machine learning results is investigated in depth with two use cases, namely model training with raw and noisy data generated by adding white Gaussian noise. For both use cases, the analysis shows where the measurement system must be improved to achieve better ML results. In general, there are two distinct possibilities, i.e., improving either the ML model or the used sensor. In case of an RMSE resulting from measurement uncertainty tending towards zero, an improvement of the ML model is suggested. In the range where U − RMSE is already very small (see Fig. 10b), a better ML model should be obtained as optimizing the sensor, including the data acquisition electronics, will only lead to even lower U − RMSE values close to zero, which does not significantly impact the overall T + U − RMSE. In contrast to that, in ranges where U − RMSE is higher, minimizing this RMSE by optimizing the physical sensor system should be the objective. To reduce the T − RMSE resulting from the ML model, using a better model would be necessary, as this can significantly influence the ML results. A better model can be achieved, for example, by using a higher number of PLSR components, as shown in this contribution, or by using deep learning, which can also improve the T − RMSE (Robin et al., 2021).

Finally, it is shown that increased robustness of the machine learning model can be achieved by adding white Gaussian noise to the raw training data.

In future work, the influence of different types of colored noise on ML results can be investigated, as this contribution has addressed only different additive white noise models. Therefore, the correlation must be considered within the uncertainty propagation, and this is only possible for the feature extractors. Furthermore, the difference between noise produced by the data acquisition electronics, especially the logarithmic amplifier as simulated in this contribution, and noise produced by the sensor could be investigated. To simulate sensor noise or electronic noise before the logarithmic amplifier noise, the noise must already be added to the inverse logarithmic of the logarithmic resistance raw data.

Appendix A: Additional figures

A1 Formaldehyde as target

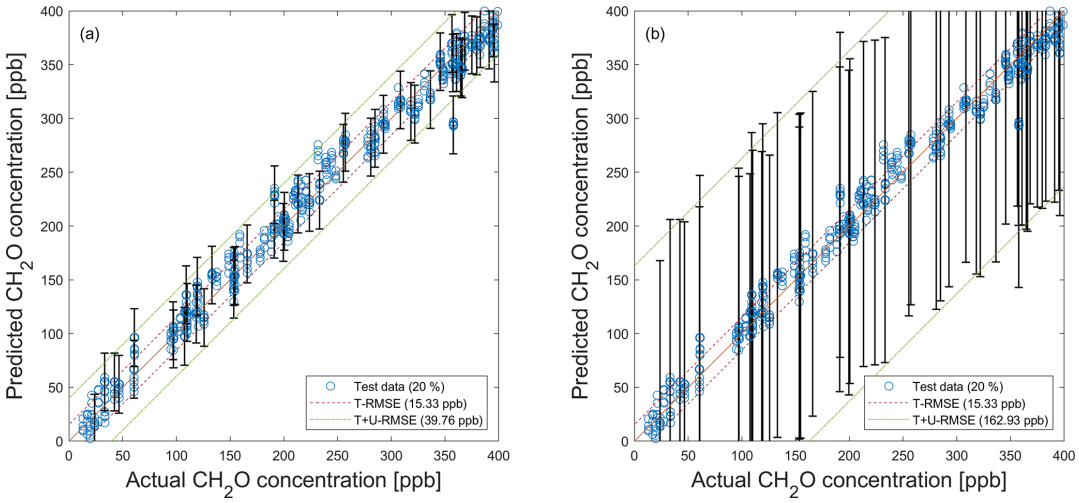


Figure A1. PLSR model (trained with raw data; $n_{\mathrm{comp}} = 20$) applied to test data (see Fig. 5) for the quantification of formaldehyde and the propagated uncertainty. **(a)** SNR = 55 dB. **(b)** SNR = 40 dB. Dashed red and green lines indicate the test RMSE (T − RMSE) and the test plus uncertainty RMSE (T + U − RMSE) based on test data, respectively. For better visibility, error bars are only shown for every 10th prediction.

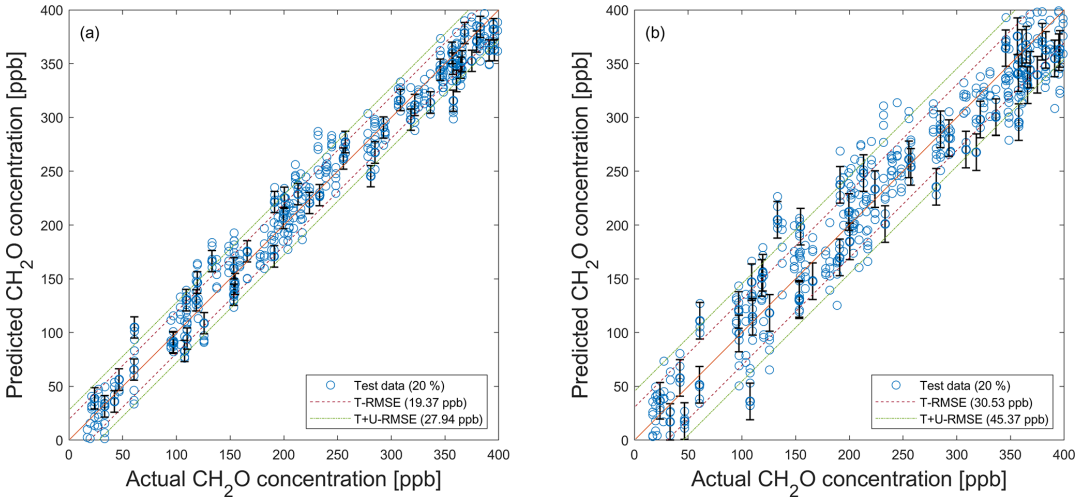


Figure A2. PLSR model (trained with noisy data; $n_{\mathrm{comp}} = 20$) applied to noisy test data (see Fig. 5) for the quantification of formaldehyde using their associated standard uncertainty. **(a)** SNR = 55 dB. **(b)** SNR = 40 dB. Dashed red and green lines indicate the test RMSE (T − RMSE) and the test plus uncertainty RMSE (T + U − RMSE) based on test data, respectively. For better visibility, error bars are only shown for every 10th prediction.

A2 VOC_{sum} as target

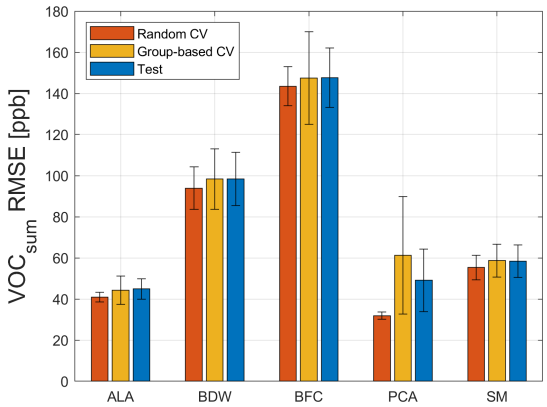


Figure A3. Random CV, group-based CV, and test RMSE of the five FE algorithms, using Pearson as FS and PLSR with $n_{comp} = 20$ for 100 trials with different data splits and the VOC_{sum} concentration as target. ALA is the adaptive linear approximation, BDW is the best Daubechies wavelets, BFC is the best Fourier coefficients, PCA is the principal component analysis, and SM is the statistical moments.

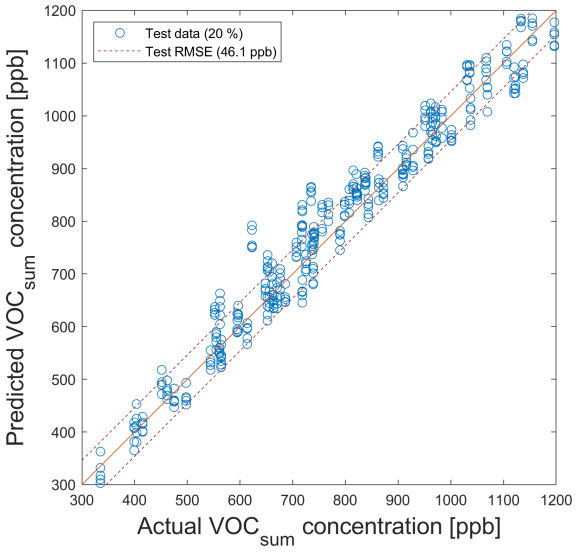


Figure A4. PLSR model for the quantification of VOC_{sum} for testing with test data from the data split shown in Fig. 5. Dashed lines indicate the RMSE of test data (T – RMSE).

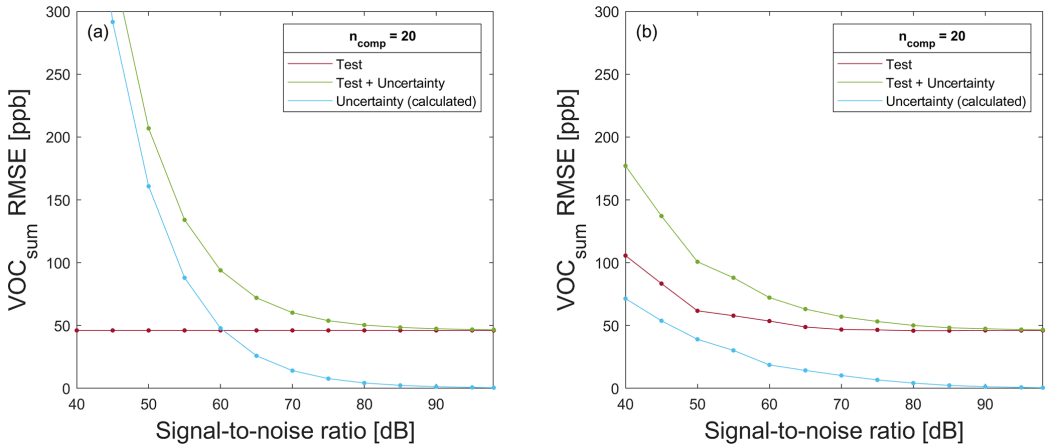


Figure A5. RMSE for testing a model trained with 80 % **(a)** raw data and **(b)** noisy data for VOC_{sum} prediction on 20 % test data without (red) and with associated uncertainty values (green), in addition to the calculated uncertainty RMSE (blue) resulting from the difference in T + U − RMSE and T − RMSE.

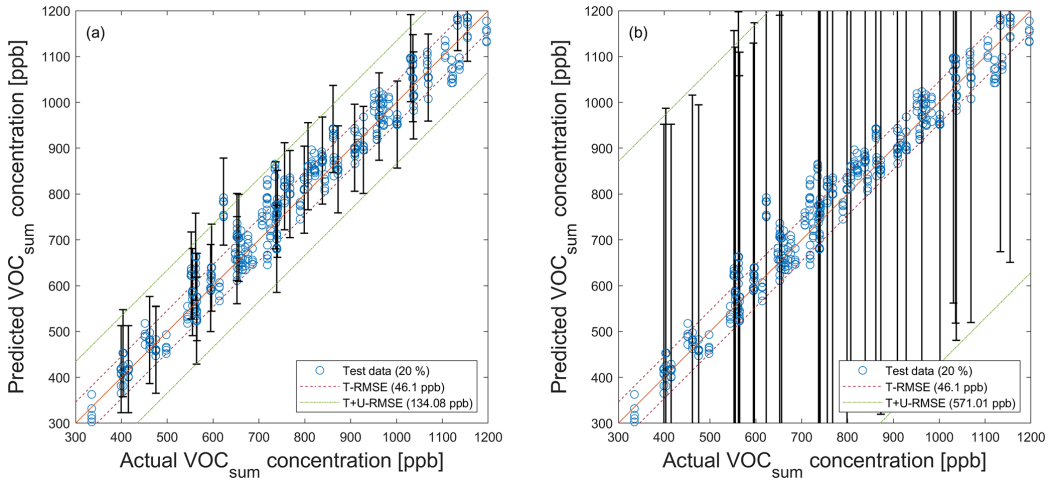


Figure A6. PLSR model (trained with raw data; $n_{comp} = 20$) applied to test data (see Fig. 5) for the quantification of VOC_{sum} and propagated uncertainty. **(a)** SNR = 55 dB. **(b)** SNR = 40 dB. Dashed red and green lines indicate the test RMSE (T − RMSE) and the test plus uncertainty RMSE (T + U − RMSE) based on test data, respectively. For better visibility, error bars are only shown for every 10th prediction.

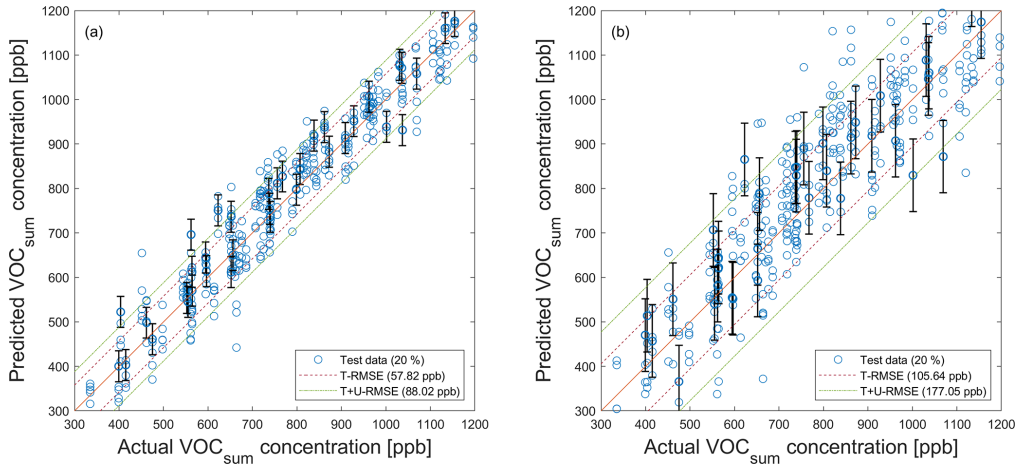


Figure A7. PLSR model (trained with noisy data; $n_{comp} = 20$) applied to noisy test data (see Fig. 5) for the quantification of VOC_{sum} using their associated standard uncertainty. **(a)** SNR = 55 dB. **(b)** SNR = 40 dB. Dashed red and green lines indicate the test RMSE (T − RMSE) and the test plus uncertainty RMSE (T + U − RMSE) based on test data, respectively. For better visibility, error bars are only shown for every 10th prediction.

A3 Additive white uniform noise and formaldehyde as target

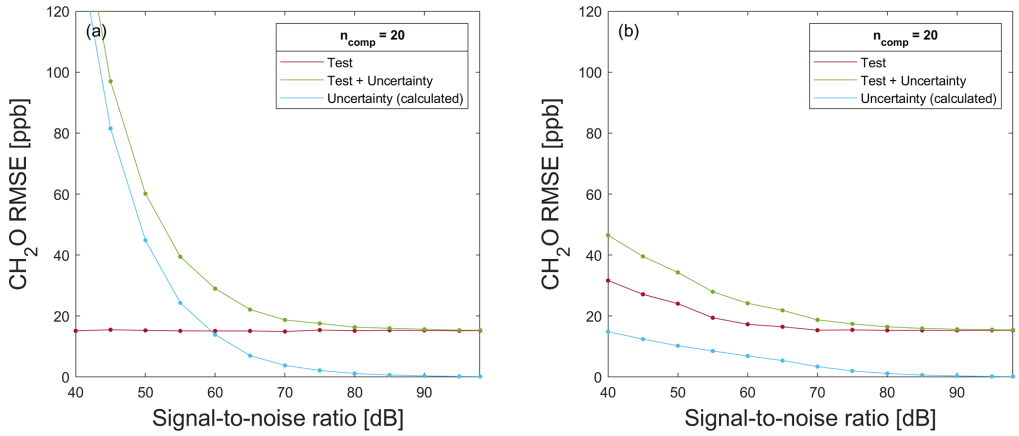


Figure A8. RMSE for testing of a model trained with 80 % **(a)** raw data and **(b)** noisy data (added white uniform noise to raw data) for a formaldehyde prediction on 20 % test data without (red) and with associated uncertainty values (green), in addition to the calculated uncertainty RMSE (blue) resulting from the difference in T + U − RMSE and T − RMSE.

Code and data availability. The paper uses data obtained from different calibration and field test measurements of gas mixtures with a MOS gas sensor. The data set is available on Zenodo https://doi.org/10.5281/zenodo.4593853 (Amann et al., 2021a).

The uncertainty-aware AMLT (Dorst et al., 2022; https://doi.org/10.1515/teme-2022-0042) includes all the code for data analysis associated with the current submission and is available at https://github.com/ZeMA-gGmbH/LMT-UA-ML-Toolbox (last access: 15 April 2022).

Author contributions. TD carried out the analysis, visualized the results, and wrote the original draft of the paper. TS, SE, and AS contributed with substantial revisions.

Competing interests. The contact author has declared that none of the authors has any competing interests.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Acknowledgements. The uncertainty-aware automated ML toolbox was developed 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. We acknowledge support by the Deutsche Forschungsgemeinschaft (DFG; German Research Foundation) and Saarland University within the "Open Access Publication Funding" program.

Financial support. This research has been supported by the European Metrology Programme for Innovation and Research (Met4FoF (grant agreement no. 17IND12)) and the European Union's Horizon 2020 Research and Innovation program.

Review statement. This paper was edited by Sebastian Wood and reviewed by two anonymous referees.

References

Amann, J., Baur, T., and Schultealbert, C.: Measuring Hydrogen in Indoor Air with a Selective Metal Oxide Semiconductor Sensor: Dataset, Zenodo [data set], https://doi.org/10.5281/zenodo.4593853, 2021a.

Amann, J., Baur, T., Schultealbert, C., and Schütze, A.: Bewertung der Innenraumluftqualität über VOC-Messungen mit Halbleitergassensoren - Kalibrierung, Feldtest, Validierung, tm - Tech. Mess., 88, S89–S94, https://doi.org/10.1515/teme-2021-0058, 2021b.

Asikainen, A., Carrer, P., Kephalopoulos, S., Fernandes, E. d. O., Wargocki, P., and Hänninen, O.: Reducing burden of disease from residential indoor air exposures in Europe (HEALTHVENT project), Environ. Health, 15, S35, https://doi.org/10.1186/s12940-016-0101-8, 2016.

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

Baur, T., Amann, J., Schultealbert, C., and Schütze, A.: Field Study of Metal Oxide Semiconductor Gas Sensors in Temperature Cycled Operation for Selective VOC Monitoring in Indoor Air, Atmosphere, 12, 647, https://doi.org/10.3390/atmos12050647, 2021.

Bennett, W. R.: Spectra of quantized signals, Bell Syst. Tech. J., 27, 446–472, https://doi.org/10.1002/j.1538-7305.1948.tb01340.x, 1948.

BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP, and OIML: JCGM 100: Evaluation of measurement data – Guide to the expression of uncertainty in measurement, https://www.bipm.org/documents/20126/2071204/JCGM_100_2008_E.pdf/cb0ef43f-baa5-11cf-3f85-4dcd86f77bd6 (last access: 18 January 2023), 2008a.

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, https://www.bipm.org/documents/20126/2071204/JCGM_101_2008_E.pdf/325dcaad-c15a-407c-1105-8b7f322d651c (last access: 18 January 2023), 2008b.

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, https://www.bipm.org/documents/20126/2071204/JCGM_102_2011_E.pdf/6a3281aa-1397-d703-d7a1-a8d58c9bf2a5 (last access: 18 January 2023), 2011.

Brasche, S. and Bischof, W.: Daily time spent indoors in German homes – Baseline data for the assessment of indoor exposure of German occupants, Int. J. Hyg. Envir. Heal., 208, 247–253, https://doi.org/10.1016/j.ijheh.2005.03.003, 2005.

Daubechies, I.: Ten Lectures on Wavelets, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, https://doi.org/10.1137/1.9781611970104, 1992.

De Jong, S.: PLS fits closer than PCR, J. Chemometr., 7, 551–557, https://doi.org/10.1002/cem.1180070608, 1993a.

De Jong, S.: SIMPLS: An alternative approach to partial least squares regression, Chemometr. Intell. Lab., 18, 251–263, https://doi.org/10.1016/0169-7439(93)85002-X, 1993b.

Dorst, T., Robin, Y., Schneider, T., and Schütze, A.: Automated ML Toolbox for Cyclic Sensor Data, MSMM 2021 – Mathematical and Statistical Methods for Metrology 2021, 149–150, http://www.msmm2021.polito.it/content/download/245/1127/file/MSMM2021_Booklet_c.pdf (last access: 18 January 2023), 2021.

Dorst, T., Schneider, T., Eichstädt, S., and Schütze, A.: Uncertainty-aware automated machine learning toolbox, tm - Tech. Mess., in press, https://doi.org/10.1515/teme-2022-0042, 2022 (code available at: https://github.com/ZeMA-gGmbH/LMT-UA-ML-Toolbox, last access: 18 January 2023).

Eicker, H.: Method and apparatus for determining the concentration of one gaseous component in a mixture of gases, US patent US4012692A, http://www.google.tl/patents/US4012692 (last access: 18 January 2023), 1977.

Ergon, R.: Principal component regression (PCR) and partial least squares regression (PLSR), John Wiley & Sons, Ltd, chap. 8, 121–142, https://doi.org/10.1002/9781118434635.ch08, 2014.

Gutierrez-Osuna, R.: Pattern analysis for machine olfaction: a review, IEEE Sens. J., 2, 189–202, https://doi.org/10.1109/JSEN.2002.800688, 2002.

Hauptmann, M., Lubin, J. H., Stewart, P. A., Hayes, R. B., and Blair, A.: Mortality from solid cancers among workers in

formaldehyde industries, Am. J. Epidemiol., 159, 1117–1130, https://doi.org/10.1093/aje/kwh174, 2004.
Horn, R. A.: The Hadamard product, in: Matrix theory and applications, edited by: Johnson, C. R., Proc. Sym. Ap., 40, 87–169, https://doi.org/10.1090/psapm/040/1059485, 1990.
Jackson, J. E.: A Use's Guide to Principal Components, John Wiley & Sons, Inc., https://doi.org/10.1002/0471725331, 1991.
Jiang, L., Djurdjanovic, D., Ni, J., and Lee, J.: Sensor Degradation Detection in Linear Systems, in: Engineering Asset Management, edited by: Mathew, J., Kennedy, J., Ma, L., Tan, A., and Anderson, D., Springer London, London, 1252–1260, https://doi.org/10.1007/978-1-84628-814-2_138, 2006.
Jones, A. P.: Indoor air quality and health, Atmos. Environ., 33, 4535–4564, https://doi.org/10.1016/S1352-2310(99)00272-1, 1999.
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, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 20–25 August 1995, IJCAI'95, 2, 1137–1143, 1995.
Lee, A. P. and Reedy, B. J.: Temperature modulation in semiconductor gas sensing, Sensor. Actuat. B-Chem., 60, 35–42, https://doi.org/10.1016/S0925-4005(99)00241-5, 1999.
Martin, H. R. and Honarvar, F.: Application of statistical moments to bearing failure detection, Appl. Acoust., 44, 67–77, https://doi.org/10.1016/0003-682X(94)P4420-B, 1995.
McKay, M. D., Beckman, R. J., and Conover, W. J.: A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code, Technometrics, 21, 239–245, https://doi.org/10.2307/1268522, 1979.
Mörchen, F.: 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.
NTP (National Toxicology Program): Report on Carcinogens, 15th edn., https://doi.org/10.22427/NTP-OTHER-1003, 2021.
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, Pittsburgh, PA, USA, https://www.cs.cmu.edu/~bobski/pubs/tr01108-twosided.pdf (last access: 18 January 2023), 2001.
Pearson, K.: 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, 559–572, https://doi.org/10.1080/14786440109462720, 1901.
Reams, R.: Hadamard inverses, square roots and products of almost semidefinite matrices, Linear Algebra Appl., 288, 35–43, https://doi.org/10.1016/S0024-3795(98)10162-3, 1999.
Robin, Y., Amann, J., Baur, T., Goodarzi, P., Schultealbert, C., Schneider, T., and Schütze, A.: High-Performance VOC Quantification for IAQ Monitoring Using Advanced Sensor Systems and Deep Learning, Atmosphere, 12, 1487, https://doi.org/10.3390/atmos12111487, 2021.
Rüffer, D., Hoehne, F., and Bühler, J.: New Digital Metal-Oxide (MOx) Sensor Platform, Sensors, 18, 1052, https://doi.org/10.3390/s18041052, 2018.
Schneider, T., Helwig, N., and Schütze, A.: Automatic feature extraction and selection for classification of cyclical time series data, tm - Tech. Mess., 84, 198–206, https://doi.org/10.1515/teme-2016-0072, 2017.
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, 2018.
Schultealbert, C., Baur, T., Schütze, A., and Sauerwald, T.: Facile Quantification and Identification Techniques for Reducing Gases over a Wide Concentration Range Using a MOS Sensor in Temperature-Cycled Operation, Sensors, 18, 744, https://doi.org/10.3390/s18030744, 2018.
Schütze, A. and Sauerwald, T.: Dynamic operation of semiconductor sensors, in: Semiconductor Gas Sensors, 2nd edn., edited by: Jaaniso, R. and Tan, O. K., Woodhead Publishing Series in Electronic and Optical Materials, Woodhead Publishing, 385–412, https://doi.org/10.1016/B978-0-08-102559-8.00012-4, 2020a.
Schütze, A. and Sauerwald, T.: Indoor air quality monitoring, in: Advanced Nanomaterials for Inexpensive Gas Microsensors, edited by: Llobet, E., Micro and Nano Technologies, Elsevier, 209–234, https://doi.org/10.1016/B978-0-12-814827-3.00011-6, 2020b.
Sensirion AG: Datasheet SGP30, https://sensirion.com/media/documents/984E0DD5/61644B8B/Sensirion_Gas_Sensors_Datasheet_SGP30.pdf (last access: 18 January 2023), 2020.
Spaul, W. A.: Building-related factors to consider in indoor air quality evaluations, J. Allergy Clin. Immun., 94, 385–389, 1994.
Sundell, J.: On the history of indoor air quality and health, Indoor air, 14, 51–58, 2004.
Thorndike, R. L.: Who belongs in the family?, Psychometrika, 18, 267–276, https://doi.org/10.1007/BF02289263, 1953.
Tsai, W.-T.: An overview of health hazards of volatile organic compounds regulated as indoor air pollutants, Rev. Environ. Health, 34, 81–89, https://doi.org/10.1515/reveh-2018-0046, 2019.
Von Pettenkofer, M.: Über den Luftwechsel in Wohngebäuden, Cotta, München, https://opacplus.bsb-muenchen.de/title/BV013009721 (last access: 18 January 2023), 1858.
Wold, S., Albano, C., Dunn, W. J., Edlund, U., Esbensen, K., Geladi, P., Hellberg, S., Johansson, E., Lindberg, W., and Sjöström, M.: Multivariate Data Analysis in Chemistry, in: Chemometrics: Mathematics and Statistics in Chemistry, edited by: Kowalski, B. R., Springer, Dordrecht, Netherlands, 17–95, https://doi.org/10.1007/978-94-017-1026-8_2, 1984.
Wold, S., Sjöström, M., and Eriksson, L.: PLS-regression: a basic tool of chemometrics, Chemometr. Intell. Lab., 58, 109–130, https://doi.org/10.1016/S0169-7439(01)00155-1, 2001.
World Health Organization (WHO): WHO guidelines for indoor air quality: selected pollutants, WHO Regional Office for Europe, Copenhagen, Vol. 9, ISBN: 978-9-2890-0213-4, 2010.
Zhang, L.: Formaldehyde, Issues in Toxicology, The Royal Society of Chemistry, https://doi.org/10.1039/9781788010269, 2018.

4 Conclusion and Outlook

With the growing digitalization in recent years and the rise of *Industry 4.0* (I4.0) as the fourth industrial revolution with digitalized factories, so-called Factories of the Future (FoFs), using the concept of *Industrial Internet of Things* (IIoT), the total amount of data increased rapidly. This large amount of data requires a high level of automation, from calibration and data collection through to data-driven analysis based on machine learning (ML). In the industrial context, the reliability of the ML results is essential, as they are used for critical processes such as fault detection, condition monitoring, and predictive maintenance. When predictions and decisions are based on data-driven ML models, confidence in the used algorithm and reliable data is required. Therefore, assessing data quality is essential in all steps of processes in FoF, beginning with the individual sensors through to the data analysis, and is one of the most important industrial needs [30]. This can be achieved by establishing a metrological framework for the entire lifecycle of measured data from traceable calibration of sensors, metrology in sensor networks, and data preprocessing through to propagation and quantification of the uncertainty in ML [31]. Combining both research fields, metrology on the one hand and ML on the other, leads to uncertainty-aware ML. As the average performance of an ML model is often only assessed by cross-validation (CV) [26, 27], the benefit of uncertainty-aware ML is that it can assess each individual prediction of an ML model.

Before performing ML, data collection and data preprocessing have to be carried out. Combining various interconnected sensors measuring the same physical quantity or different physical quantities in a distributed sensor network is the key concept in IIoT. Thus, sensor data fusion is one typical challenge in distributed sensor networks in an FoF. Therefore, a data pipeline for time series data has been introduced to align data of several sensors of two different data acquisition units (DAQs), one low-cost and one high-cost system, whose only connection is a trigger signal. As a measurement result is only complete when it contains a measured value and the accompanied measurement uncertainty, the raw data's measurement uncertainty has been calculated using dynamic calibration information. As cost awareness is an important topic in industry and economy,

it is shown that using only the low-cost DAQ data for ML leads to a good trade-off between cost and accuracy. In the context of the presented sensor data fusion process, a new data set has been published, which fulfills most of the requirements of the FAIR data principles. Evaluating the data set according to the FAIR data maturity model results in good agreement with the FAIR data principles.

When predictions and decisions are based on data, confidence in the used algorithm and reliable data is required. A further topic of this dissertation is research on the effects of data quality on ML performance. Measurement uncertainty, as an indicator for data quality, occurs in time and value and allows to evaluate decisions made by ML models. By introducing generated artificial time shifts on a raw data set, time synchronization errors between individual sensors' cycles have been studied. Already time shifts of 0.1 ms in the data lead to poor performance of the ML model. Thus, further investigations have been carried out concerning enhancing the ML model performance in the presence of time synchronization problems. Omitting phases out of the feature set has been proven to increase ML model performance. However, the best results have been achieved by adding data with artificial time shifts of different values in the training data. This technique is already well known as data augmentation in deep learning (DL) to increase the amount of data used to train neural networks and avoid overfitting [234, 235].

As the evaluation of measurement uncertainty for each specific prediction is often neglected in ML, this dissertation addresses this topic. The basis for investigating the influence of measurement uncertainty in the value on ML results is an already existing and published automated machine learning toolbox (AMLT) for classification problems. This AMLT has been extended to an uncertainty-aware version named uncertainty-aware automated machine learning toolbox (UA-AMLT) by developing uncertainty propagation for the feature extraction (FE) algorithms as well as for the classification carried out using *Linear Discriminant Analysis* (LDA). For the uncertainty propagation through these algorithms, analytical approaches have been presented in line with the *Guide to the Expression of Uncertainty in Measurement* (GUM) and its supplements, Supplement 1 (GUM-S1) and Supplement 2 (GUM-S2). As feature selection (FS) only selects features, the selection process is made uncertainty-aware by introducing modified versions of each FS algorithm, i.e., weighted FS algorithm, that uses the uncertainty values as weights. Moreover, this toolbox was made applicable for quantification by introducing an uncertainty-aware *Partial Least Squares Regression* (PLSR) algorithm based on uncertainty propagation in line with GUM-S2.

The investigation of the influence of measurement uncertainty on ML performance has been demonstrated for a data set consisting of raw sensor data of a smart gas sensor. To simulate measurement uncertainty in the data, noise as one source of measurement uncertainty is added to the raw sensor data. In this dissertation, artificially generated white Gaussian noise (a special kind of noise with zero mean and finite variance) of different signal-to-noise ratio (SNR) levels is used. Investigating the influence of measurement uncertainty on ML results consists of two approaches: model training with raw data and model training with noisy data. Using the suggested approaches, statements about where the overall system needs to be enhanced are possible. This can be either the used sensor and measuring electronic or the trained ML model. Using a sensor with higher precision and optimizing the data acquisition electronics is necessary in case the root-mean-square error (RMSE) contribution of the uncertainty values is high. In contrast, improving the trained model by, e.g., tuning hyperparameters of an ML model, must be carried out when the RMSE contribution of the uncertainty values already tends towards zero, and only the RMSE contribution of the model can significantly influence the overall RMSE. It has also been shown that training with noisy data increases the robustness of the ML model against random noise. As energy efficiency and energy saving are currently important topics in industry, computational cost can be reduced using the UA-AMLT because a model must only be trained once with raw data and the measurement uncertainty, even if changing significantly during the process, can be propagated through the model.

The topics presented in this dissertation have great potential to improve the confidence of ML decisions, which makes them worth further investigation. First, the UA-AMLT for regression problems can be improved by implementing additional FS algorithms so that not only weighted Pearson correlation must be used. For example, Regressional ReliefF (RReliefF) [92, 236], an algorithm similar to ReliefF but suitable for quantification problems, is proposed for the toolbox for regression problems. Instead of *Recursive Feature Elimination Support Vector Machine* (RFESVM), which is only suitable for classification problems, other recursive feature elimination algorithms such as *Recursive Feature Elimination Support Vector Regression* (RFESVR) and *Recursive Feature Elimination Least Squares Regression* (RFELSR) [237] can be implemented. All these suggested algorithms need to be made uncertainty-aware to add them to the UA-AMLT.

Moreover, further analysis can be carried out with the UA-AMLT. The influence of different types of colored noise on ML performance is of interest here. To use colored

noise in the UA-AMLT, algorithms that can also deal with correlated input quantities are required. This is yet only implemented for the FE algorithms. Thus, the FS algorithms, as well as LDA for classification and PLSR for quantification, must be adapted for correlated input quantities.

This dissertation only addresses supervised ML techniques in the UA-AMLT. Thus, investigating the influence of measurement uncertainty on novelty detection as a semi-supervised ML technique or clustering methods as unsupervised ML algorithms are suggested.

For future work, it is interesting to transfer the results achieved by investigating the influence of measurement uncertainty on ML performance using data of a smart gas sensor to applications in industry, e.g., remaining useful lifetime (RUL) estimations. The knowledge of where the measurement system, including the ML process, must be improved (either the sensor/electronics or the ML model) to obtain better ML results can thus be transferred to industrial applications leading to more reliable ML predictions. As ML results can be improved using DL instead, analysis of measurement uncertainty influence on results based on DL models is also worth investigating in future work.

References

[1] BIPM et al. *JCGM 200: International Vocabulary of Metrology - Basic and General Concepts and Associated Terms (VIM)*. 2012. URL: https://www.bipm.org/documents/20126/2071204/JCGM_200_2012.pdf/f0e1ad45-d337-bbeb-53a6-15fe649d0ff1.

[2] Horst Czichos, Tetsuya Saito, and Leslie Smith, eds. *Springer Handbook of Metrology and Testing*. 2nd ed. Berlin + Heidelberg, Germany: Springer, 2011. ISBN: 978-3-642-16641-9. DOI: 10.1007/978-3-642-16641-9.

[3] René Just Haüy. *Instruction abrégée sur les mesures déduites de la grandeur de la terre, uniformes pour toute la République, et sur les calculs relatifs à leur division décimale; par la Commission temporaire des poids & mesures républicaines, en exécution des décrets de la Convention nationale*. Edition originale. Paris, France: De l'Imprimerie nationale exécutive du Louvre, 1793. URL: https://archive.org/details/instructionabreg00hauy.

[4] BIPM. *Le Système international d'unités (SI)*. 9th ed. 2019. URL: https://www.bipm.org/documents/20126/41483022/SI-Brochure-9.pdf.

[5] Première République française. "Loi du 18 germinal an III relative aux poids et mesures." In: *Bulletin des lois de la République française*. Vol. 1. 135. 1795, pp. 1–8.

[6] Miguel A. Martin-Delgado. "The new SI and the fundamental constants of nature." In: *European Journal of Physics* 41.6 (2020), p. 63003. DOI: 10.1088/1361-6404/abab5e.

[7] BIPM et al. *JCGM 100: Evaluation of measurement data – Guide to the expression of uncertainty in measurement*. 2008. URL: https://www.bipm.org/documents/20126/2071204/JCGM_100_2008_E.pdf/cb0ef43f-baa5-11cf-3f85-4dcd86f77bd6.

[8] Tahera Kalsoom et al. "Advances in Sensor Technologies in the Era of Smart Factory and Industry 4.0." In: *Sensors* 20.23 (Nov. 2020), p. 6783. ISSN: 1424-8220. DOI: 10.3390/s20236783.

[9] Andreas Schütze, Nikolai Helwig, and Tizian Schneider. "Sensors 4.0 - Smart sensors and measurement technology enable Industry 4.0." In: *Journal of Sensors and Sensor Systems* 7.1 (2018), pp. 359–371. ISSN: 2194-878X. DOI: 10.5194/jsss-7-359-2018.

[10] Jyotir Moy Chatterjee, Harish Garg, and R. N. Thakur, eds. *A Roadmap for Enabling Industry 4.0 by Artificial Intelligence*. John Wiley & Sons, Inc. and Scrivener Publishing LLC, 2023. ISBN: 978-1-119-90485-4.

[11] Tencent Research Institute et al., eds. *Artificial Intelligence*. Singapore, Singapore: Springer, 2021. ISBN: 978-981-15-6547-2. DOI: 10.1007/978-981-15-6548-9.

[12] Levity AI GmbH. *Deep Learning vs. Machine Learning – What's The Difference?* URL: https://levity.ai/blog/difference-machine-learning-deep-learning. [Online; accessed December 12, 2022].

[13] James Moor. "The Dartmouth College Artificial Intelligence Conference: The Next Fifty Years." In: *AI Magazine* 27.4 (2006), p. 87. DOI: 10.1609/aimag.v27i4.1911.

[14] Wolfgang Ertel. *Introduction to Artificial Intelligence*. 2nd ed. Cham, Switzerland: Springer, 2017. ISBN: 978-3-319-58487-4. DOI: 10.1007/978-3-319-58487-4.

[15] Ameet V. Joshi. *Machine Learning and Artificial Intelligence*. Cham, Switzerland: Springer, 2020. ISBN: 978-3-030-26622-6. DOI: 10.1007/978-3-030-26622-6.

[16] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. 4th ed. Pearson, 2022. ISBN: 978-1-292-40113-3.

[17] Pei Wang. "On Defining Artificial Intelligence." In: *Journal of Artificial General Intelligence* 10.2 (2019), pp. 1–37. DOI: 10.2478/jagi-2019-0002.

[18] Arthur L. Samuel. "Some Studies in Machine Learning Using the Game of Checkers." In: *IBM Journal of Research and Development* 3.3 (1959), pp. 210–229. DOI: 10.1147/rd.33.0210.

[19] Tong Meng et al. "A survey on machine learning for data fusion." In: *Information Fusion* 57 (May 2020), pp. 115–129. ISSN: 1566-2535. DOI: 10.1016/j.inffus.2019.12.001.

[20] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." In: *Nature* 521.7553 (2015), pp. 436–444. ISSN: 1476-4687. DOI: 10.1038/nature14539.

[21] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. Adaptive Computation and Machine Learning series. Cambridge, MA, USA: MIT Press, 2016. ISBN: 978-0-262-03561-3.

[22] Christian Janiesch, Patrick Zschech, and Kai Heinrich. "Machine learning and deep learning." In: *Electronic Markets* 31.3 (2021), pp. 685–695. ISSN: 1422-8890. DOI: 10.1007/s12525-021-00475-2.

[23] Jörg Frochte. *Maschinelles Lernen*. 2nd ed. Hanser, 2019. ISBN: 978-3-446-45996-0.

[24] Charu C. Aggarwal. *Neural Networks and Deep Learning*. Cham, Switzerland: Springer, 2018. ISBN: 978-3-319-94463-0. DOI: 10.1007/978-3-319-94463-0.

[25] Cynthia Rudin. "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead." In: *Nature Machine Intelligence* 1.5 (2019), pp. 206–215. ISSN: 2522-5839. DOI: 10.1038/s42256-019-0048-x.

[26] Francesca Tavazza, Brian DeCost, and Kamal Choudhary. "Uncertainty Prediction for Machine Learning Models of Material Properties." In: *ACS Omega* 6.48 (Dec. 2021), pp. 32431–32440. DOI: 10.1021/acsomega.1c03752.

[27] Gitte Vanwinckelen and Hendrik Blockeel. "On Estimating Model Accuracy with Repeated Cross-Validation." In: *BeneLearn 2012: Proceedings of the 21st Belgian-Dutch Conference on Machine Learning* (Ghent, Belgium, May 24–25, 2012). 2012, pp. 39–44. ISBN: 978-94-6197-044-2.

[28] 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* (Montreal, Quebec, Canada, Aug. 20–25, 1995). IJCAI'95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995, 1137—1143. ISBN: 978-1-55860-363-9.

[29] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. New York, NY, USA: Springer, 2009. ISBN: 978-0-387-84858-7. DOI: 10.1007/978-0-387-84858-7.

[30] Tanja Dorst et al. “Metrology for the factory of the future: Towards a case study in condition monitoring.” In: *2019 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)* (Auckland, New Zealand, May 20–23, 2019). 2019, pp. 439–443. ISBN: 978-1-5386-3460-8. DOI: 10.1109/I2MTC.2019.8826973.

[31] Sascha Eichstädt. *Final Publishable Report for 17IND12 Met4FoF “Metrology for the Factory of the Future”*. Version Final. Nov. 2021. DOI: 10.5281/zenodo.6620849.

[32] Anupam P. Vedurmudi et al. “Semantic Description of Quality of Data in Sensor Networks.” In: *Sensors* 21.19 (2021). ISSN: 1424-8220. DOI: 10.3390/s21196462.

[33] Jean-Laurent Hippolyte et al. “Using Ontologies to Create Machine-Actionable Datasets: Two Case Studies.” In: *Metrology* 3.1 (Feb. 2023), pp. 65–80. ISSN: 2673-8244. DOI: 10.3390/metrology3010003.

[34] QUDT.org. *QUDT CATALOG - Quantities, Units, Dimensions and Data Types Ontologies*. Tech. rep. 2022. URL: http://www.qudt.org/2.1/catalog/qudt-catalog.html.

[35] W3C. *Semantic Sensor Network Ontology*. Tech. rep. 2021. URL: https://www.w3.org/TR/vocab-ssn/.

[36] Benedikt Seeger and Thomas Bruns. “Primary calibration of mechanical sensors with digital output for dynamic applications.” In: *Acta IMEKO* 10.3, article 24 (Sept. 2021), pp. 177–184. DOI: 10.21014/acta_imeko.v10i3.1075.

[37] Tizian Schneider, Nikolai Helwig, and Andreas Schütze. “Industrial condition monitoring with smart sensors using automated feature extraction and selection.” In: *Measurement Science and Technology* 29.9 (2018). ISSN: 1361-6501. DOI: 10.1088/1361-6501/aad1d4.

[38] Tanja Dorst et al. “Automated ML Toolbox for Cyclic Sensor Data.” In: *MSMM 2021 - Mathematical and Statistical Methods for Metrology* (Online, May 31–June 1, 2021). 2021, pp. 149–150. URL: http://www.msmm2021.polito.it/content/download/245/1127/file/MSMM2021_Booklet_c.pdf.

[39] Tizian Schneider, Nikolai Helwig, and Andreas Schütze. “Automatic feature extraction and selection for condition monitoring and related datasets.” In: *2018 IEEE International Instrumentation and Measurement Technology Conference*

(I2MTC) (Houston, TX, USA, May 14–17, 2018). 2018, pp. 429–434. ISBN: 978-1-5386-2222-3. DOI: 10.1109/I2MTC.2018.8409763.

[40] Sascha Eichstädt and Volker Wilkens. "GUM2DFT —a software tool for uncertainty evaluation of transient signals in the frequency domain." In: *Measurement Science and Technology* 27.5 (2016), p. 055001. DOI: 10.1088/0957-0233/27/5/055001.

[41] Lorenzo Peretto, Renato Sasdelli, and Roberto Tinarelli. "Uncertainty Propagation in the Discrete-Time Wavelet Transform." In: *IEEE Transactions on Instrumentation and Measurement* 54.6 (2005), pp. 2474–2480. ISSN: 1557-9662. DOI: 10.1109/TIM.2005.858145.

[42] Lorenzo Peretto, Renato Sasdelli, and Roberto Tinarelli. "On uncertainty in wavelet-based signal analysis." In: *IEEE Transactions on Instrumentation and Measurement* 54.4 (2005), pp. 1593–1599. ISSN: 0018-9456. DOI: 10.1109/TIM.2005.851210.

[43] BIPM et al. *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. URL: https://www.bipm.org/documents/20126/2071204/JCGM_101_2008_E.pdf/325dcaad-c15a-407c-1105-8b7f322d651c.

[44] BIPM et al. *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. URL: https://www.bipm.org/documents/20126/2071204/JCGM_102_2011_E.pdf/6a3281aa-1397-d703-d7a1-a8d58c9bf2a5.

[45] Sheeba Samuel, Frank Löffler, and Birgitta König-Ries. "Machine Learning Pipelines: Provenance, Reproducibility and FAIR Data Principles." In: *Provenance and Annotation of Data and Processes*. Ed. by Boris Glavic, Vanessa Braganholo, and David Koop. Cham, Switzerland: Springer, 2021, pp. 226–230. ISBN: 978-3-030-80960-7.

[46] Iqbal H. Sarker. "Machine Learning: Algorithms, Real-World Applications and Research Directions." In: *SN Computer Science* 2.3 (2021), p. 160. ISSN: 2661-8907. DOI: 10.1007/s42979-021-00592-x.

[47] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining*. 3rd ed. The Morgan Kaufmann Series in Data Management Systems. Boston, MA, USA: Morgan Kaufmann, 2012. ISBN: 978-0-12-381479-1. DOI: 10.1016/C2009-0-61819-5.

[48] Christian Böhm, Stefan Berchtold, and Daniel A. Keim. "Searching in High-Dimensional Spaces: Index Structures for Improving the Performance of Multimedia Databases." In: *ACM Computing Surveys* 33.3 (Sept. 2001), pp. 322–373. ISSN: 0360-0300. DOI: 10.1145/502807.502809.

[49] Richard Bellman. *Dynamic programming*. 3rd ed. Princeton, NJ, USA: Princeton University Press, 1957. ISBN: 0-691-07951-X.

[50] Kevin Beyer et al. "When Is "Nearest Neighbor" Meaningful?" 7th International Conference on Database Theory (ICDT). In: *Database Theory - ICDT'99* (Jerusalem, Israel, Jan. 10–12, 1999). Berlin + Heidelberg, Germany: Springer, 1999, pp. 217–235. ISBN: 978-3-540-49257-3.

[51] Ricardo Gutierrez-Osuna. "Pattern analysis for machine olfaction: a review." In: *IEEE Sensors Journal* 2.3 (2002), pp. 189–202. DOI: 10.1109/JSEN.2002.800688.

[52] Michel Verleysen and Damien François. "The Curse of Dimensionality in Data Mining and Time Series Prediction." 8th International Work-Conference on Artificial Neural Networks (IWANN). In: *Computational Intelligence and Bioinspired Systems* (Barcelona, Spain, June 8–10, 2005). Ed. by Joan Cabestany, Alberto Prieto, and Francisco Sandoval. Berlin + Heidelberg, Germany: Springer, 2005, pp. 758–770. ISBN: 978-3-540-32106-4.

[53] Muhammad Habib ur Rehman et al. "Big Data Reduction Methods: A Survey." In: *Data Science and Engineering* 1.4 (2016), pp. 265–284. ISSN: 2364-1541. DOI: 10.1007/s41019-016-0022-0.

[54] David H. Wolpert and William G. Macready. "No Free Lunch Theorems for Optimization." In: *IEEE Transactions on Evolutionary Computation* 1.1 (1997), pp. 67–82. DOI: 10.1109/4235.585893.

[55] David H. Wolpert. "The Supervised Learning No-Free-Lunch Theorems." In: *Soft Computing and Industry: Recent Applications*. Ed. by Rajkumar Roy et al. London, United Kingdom: Springer, 2002, pp. 25–42. ISBN: 978-1-4471-0123-9. DOI: 10.1007/978-1-4471-0123-9_3.

[56] Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*. New York, NY, USA: Springer, 2013. ISBN: 978-1-4614-6848-6. DOI: 10.1007/978-1-4614-6849-3.

[57] Tizian Schneider. "Methoden der automatisierten Merkmalextraktion und -selektion von Sensorsignalen." Master thesis. Naturwissenschaftlich-Technische Fakultät II, Saarland University, Saarbruecken, 2015.

[58] Tanja Dorst et al. "Influence of measurement uncertainty on machine learning results demonstrated for a smart gas sensor." In: *Journal of Sensors and Sensor Systems* 12.1 (2023), pp. 45–60. DOI: 10.5194/jsss-12-45-2023.

[59] Robert T. Olszewski. "Generalized feature extraction for structural pattern recognition in time-series data." PhD thesis. Pittsburgh, PA, USA: Carnegie Mellon University, Feb. 2001. ISBN: 978-0-493-53871-6.

[60] He-Jun Jiao et al. "Applications of wavelet analysis to cloud computing and big data: Status and prospects." In: *2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)* (Chengdu, China, Dec. 16–18, 2016). 2016, pp. 127–130. DOI: 10.1109/ICCWAMTIP.2016.8079820.

[61] Joshua Caleb Dagadu et al. "DWT based encryption technique for medical images." In: *2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)* (Chengdu, China, Dec. 16–18, 2016). 2016, pp. 252–255. DOI: 10.1109/ICCWAMTIP.2016.8079849.

[62] Tatyana N. Kruglova. "Wavelet analysis for fault diagnosis of electrical machines using current signals." In: *2016 2nd International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)* (Chelyabinsk, Russia, May 19–20, 2016). 2016, pp. 1–5. DOI: 10.1109/ICIEAM.2016.7911652.

[63] Dengsheng Zhang. "Wavelet Transform." In: *Fundamentals of Image Data Mining: Analysis, Features, Classification and Retrieval*. Cham, Switzerland: Springer, 2019, pp. 35–44. ISBN: 978-3-030-17989-2. DOI: 10.1007/978-3-030-17989-2_3.

[64] Dennis Gabor. "Theory of communication." In: *Journal of the Institution of Electrical Engineering* 93.26 (Nov. 1946), pp. 429–457. ISSN: 0367-7540.

[65] Mani Mehra. *Wavelets Theory and Its Applications*. Singapore, Singapore: Springer, 2018. ISBN: 978-981-13-2594-6. DOI: 10.1007/978-981-13-2595-3.

[66] Martin Vetterli and Cormac Herley. "Wavelets and Filter Banks: Theory and Design." In: *IEEE Transactions on Signal Processing* 40 (1992), pp. 2207–2232. DOI: 10.1109/78.157221.

[67] Ingrid Daubechies. *Ten Lectures on Wavelets*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1992. ISBN: 978-0-89871-274-2.

[68] Fabian Mörchen. "Time series feature extraction for data mining using DWT and DFT." In: *Department of Mathematics and Computer Science, University of Marburg, Germany - Technical Report* 33 (2003), pp. 1–31.

[69] James W. Cooley and John W. Tukey. "An Algorithm for the Machine Calculation of Complex Fourier Series." In: *Mathematics of Computation* 19 (1965), pp. 297–301.

[70] William L. Briggs and Van Emden Henson. *The DFT: An Owner's Manual for the Discrete Fourier Transform*. Other Titles in Applied Mathematics. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics (SIAM), 1995. ISBN: 978-0-89871-342-8. DOI: 10.1137/1.9781611971514.

[71] Karl Pearson. "On lines and planes of closest fit to systems of points in space." In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), pp. 559–572. DOI: 10.1080/14786440109462720.

[72] Harold Hotelling. "Analysis of a complex of statistical variables into principal components." In: *Journal of Educational Psychology* 24.6 (1933), pp. 417–441. DOI: 10.1037/h0071325.

[73] Svante Wold, Kim Esbensen, and Paul Geladi. "Principal Component Analysis." In: *Chemometrics and Intelligent Laboratory Systems* 2.1-3 (1987). Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists, pp. 37–52. DOI: 10.1016/0169-7439(87)80084-9.

[74] J. Edward Jackson. *A Use's Guide to Principal Components*. John Wiley & Sons, Inc., 1991. ISBN: 978-0-47162-267-3. DOI: 10.1002/0471725331.

[75] The MathWorks Inc. *Principal component analysis of raw data*. URL: https://de.mathworks.com/help/stats/pca.html. [Online; accessed November 3, 2022].

[76] Hugh R. Martin and Farhang Honarvar. "Application of statistical moments to bearing failure detection." In: *Applied Acoustics* 44.1 (1995), pp. 67–77. ISSN: 0003-682X. DOI: 10.1016/0003-682X(94)P4420-B.

[77] Nikolai Helwig, Eliseo Pignanelli, and Andreas Schütze. "Detecting and Compensating Sensor Faults in a Hydraulic Condition Monitoring System." In: *SENSOR 2015 Proceedings* (Nürnberg, Germany, May 19–21, 2015). 2015, pp. 641–646. DOI: 10.5162/sensor2015/D8.1.

[78] Tilo Arens et al. *Mathematik*. 5th ed. Springer eBook Collection. Berlin + Heidelberg, Germany: Springer Spektrum, 2022. ISBN: 978-3-662-64389-1. DOI: 10.1007/978-3-662-64389-1.

[79] Jundong Li et al. "Feature Selection: A Data Perspective." In: *ACM Computing Surveys* 50.6 (2017). ISSN: 0360-0300. DOI: 10.1145/3136625.

[80] B. Venkatesh and J. Anuradha. "A Review of Feature Selection and Its Methods." In: *Cybernetics and Information Technologies* 19.1 (2019), pp. 3–26. DOI: 10.2478/cait-2019-0001.

[81] Douglas M. Hawkins. "The Problem of Overfitting." In: *Journal of Chemical Information and Computer Sciences* 44.1 (Jan. 2004), pp. 1–12. ISSN: 0095-2338. DOI: 10.1021/ci0342472.

[82] Robert Tibshirani. "Regression Shrinkage and Selection via the Lasso." In: *Journal of the Royal Statistical Society. Series B* 58.1 (1996), pp. 267–288.

[83] Arthur E. Hoerl and Robert W. Kennard. "Ridge Regression: Biased Estimation for Nonorthogonal Problems." In: *Technometrics* 12.1 (1970), pp. 55–67. DOI: 10.1080/00401706.1970.10488634.

[84] Arthur E. Hoerl and Robert W. Kennard. "Ridge Regression: Applications to Nonorthogonal Problems." In: *Technometrics* 12.1 (1970), pp. 69–82. DOI: 10.1080/00401706.1970.10488635.

[85] Hui Zou and Trevor Hastie. "Regularization and Variable Selection via the Elastic Net." In: *Journal of the Royal Statistical Society. Series B* 67.2 (2005), pp. 301–320.

[86] Louis L. Thurstone. "A method of calculating the Pearson correlation coefficient without the use of deviations." In: *Psychological Bulletin* 14.1 (1917), pp. 28–32. DOI: 10.1037/h0074202.

[87] Jacob Benesty et al. "Pearson Correlation Coefficient." In: *Noise Reduction in Speech Processing*. Berlin + Heidelberg, Germany: Springer, 2009, pp. 1–4. ISBN: 978-3-642-00296-0. DOI: 10.1007/978-3-642-00296-0_5.

[88] Kenji Kira and Larry A. Rendell. "The Feature Selection Problem: Traditional Methods and a New Algorithm." In: *Proceedings of the Tenth National Conference on Artificial Intelligence* (San Jose, CA, USA, July 12–16, 1992). AAAI Press, 1992, pp. 129–134. ISBN: 978-0-262-51063-4.

[89] Kenji Kira and Larry A. Rendell. "A Practical Approach to Feature Selection." In: *Machine Learning Proceedings*. Ed. by Derek Sleeman and Peter Edwards. San Francisco, CA, USA: Morgan Kaufmann, 1992, pp. 249–256. ISBN: 978-1-55860-247-2. DOI: 10.1016/B978-1-55860-247-2.50037-1.

[90] Igor Kononenko and Se June Hong. "Attribute selection for modelling." In: *Future Generation Computer Systems* 13.2-3 (Nov. 1997), pp. 181–195. ISSN: 0167-739X. DOI: 10.1016/S0167-739X(97)81974-7.

[91] Igor Kononenko, Edvard Šimec, and Marko Robnik-Šikonja. "Overcoming the Myopia of Inductive Learning Algorithms with RELIEFF." In: *Applied Intelligence* 7.1 (Jan. 1997), pp. 39–55. ISSN: 1573-7497. DOI: 10.1023/A:1008280620621.

[92] Marko Robnik-Šikonja and Igor Kononenko. "Theoretical and Empirical Analysis of ReliefF and RReliefF." In: *Machine Learning* 53.1 (2003), pp. 23–69. ISSN: 1573-0565. DOI: 10.1023/A:1025667309714.

[93] Ryan J. Urbanowicz et al. "Relief-based feature selection: Introduction and review." In: *Journal of Biomedical Informatics* 85 (2018), pp. 189–203. ISSN: 1532-0464. DOI: 10.1016/j.jbi.2018.07.014.

[94] Isabelle Guyon et al. "Gene Selection for Cancer Classification using Support Vector Machines." In: *Machine Learning* 46.1 (2002), pp. 389–422. ISSN: 1573-0565. DOI: 10.1023/A:1012487302797.

[95] Alain Rakotomamonjy. "Variable Selection Using SVM-based Criteria." In: *Journal of Machine Learning Research* 3 (Mar. 2003), pp. 1357–1370. ISSN: 1532-4435.

[96] Yvan Saeys, Iñaki Inza, and Pedro Larrañaga. "A review of feature selection techniques in bioinformatics." In: *Bioinformatics* 23.19 (2007), pp. 2507–2517. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btm344.

[97] Hector Sanz et al. "SVM-RFE: selection and visualization of the most relevant features through non-linear kernels." In: *BMC Bioinformatics* 19.1 (2018), p. 432. ISSN: 1471-2105. DOI: 10.1186/s12859-018-2451-4.

[98] Erin L. Allwein, Robert E. Schapire, and Yoram Singer. "Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers." In: *Journal of Machine Learning Research* 1 (Dec. 2000), pp. 113–141. ISSN: 1532-4435.

[99] Corinna Cortes and Vladimir Vapnik. "Support-vector networks." In: *Machine Learning* 20.3 (1995), pp. 273–297. ISSN: 1573-0565. DOI: 10.1007/BF00994018.

[100] Isabelle Guyon and André Elisseeff. "An Introduction to Variable and Feature Selection." In: *Journal of Machine Learning Research* 3 (Mar. 2003), pp. 1157–1182. ISSN: 1532-4435.

[101] Anil K. Jain, M. Narasimha Murty, and Patrick J. Flynn. "Data Clustering: A Review." In: *ACM Computing Surveys* 31.3 (Sept. 1999), pp. 264–323. ISSN: 0360-0300. DOI: 10.1145/331499.331504.

[102] Absalom E. Ezugwu et al. "A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects." In: *Engineering Applications of Artificial Intelligence* 110 (2022), p. 104743. ISSN: 0952-1976. DOI: 10.1016/j.engappai.2022.104743.

[103] Christopher J. C. H. Watkins and Peter Dayan. "Q-learning." In: *Machine Learning* 8.3 (1992), pp. 279–292. ISSN: 1573-0565. DOI: 10.1007/BF00992698.

[104] Marco Wiering and Martijn Van Otterlo, eds. *Reinforcement Learning*. Berlin + Heidelberg, Germany: Springer, 2012. ISBN: 978-3-642-27645-3. DOI: 10.1007/978-3-642-27645-3.

[105] OpenAI. *Part 2: Kinds of RL Algorithms*. 2018. URL: https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html. [Online; accessed February 17, 2022].

[106] Markos Markou and Sameer Singh. "Novelty detection: a review—part 1: statistical approaches." In: *Signal Processing* 83.12 (2003), pp. 2481–2497. ISSN: 0165-1684. DOI: 10.1016/j.sigpro.2003.07.018.

[107] Marco A. F. Pimentel et al. "A review of novelty detection." In: *Signal Processing* 99 (2014), pp. 215–249. ISSN: 0165-1684. DOI: 10.1016/j.sigpro.2013.12.026.

[108] Gordon F. Hughes. "On the mean accuracy of statistical pattern recognizers." In: *IEEE Transactions on Information Theory* 14.1 (1968), pp. 55–63. DOI: 10.1109/TIT.1968.1054102.

[109] Ronald Aylmer Fisher. "The use of multiple measurements in taxonomic problems." In: *Annals of Eugenics* 7.2 (Sept. 1936), pp. 179–188. DOI: 10.1111/j.1469-1809.1936.tb02137.x.

[110] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. 2nd ed. A Wiley-Interscience publication. New York, NY, USA: John Wiley & Sons, Inc., 2001. ISBN: 978-0-471-05669-0.

[111] Dijun Luo, Chris Ding, and Heng Huang. "Linear Discriminant Analysis: New Formulations and Overfit Analysis." In: *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence* (San Francisco, CA, USA, Aug. 7–11, 2011). Vol. 25. 1. 2011, pp. 417–422. DOI: 10.1609/aaai.v25i1.7926.

[112] Klaus Backhaus et al. *Multivariate Analysemethoden: Eine anwendungsorientierte Einführung*. Berlin + Heidelberg, Germany: Springer Gabler, 2018. ISBN: 978-3-662-56655-8. DOI: 10.1007/978-3-662-56655-8.

[113] Alaa Tharwat et al. "Linear discriminant analysis: A detailed tutorial." In: *AI Communications* 30 (2017), pp. 169–190. ISSN: 1875-8452. DOI: 10.3233/AIC-170729.

[114] Tao Li, Shenghuo Zhu, and Mitsunori Ogihara. "Using discriminant analysis for multi-class classification: an experimental investigation." In: *Knowledge and Information Systems* 10.4 (2006), pp. 453–472. ISSN: 0219-3116. DOI: 10.1007/s10115-006-0013-y.

[115] Prasanta Chandra Mahalanobis. "On tests and measures of group divergence." In: *Journal of the Asiatic Society of Bengal* 26 (1930), pp. 541–588.

[116] Prasanta Chandra Mahalanobis. "On the generalized distance in statistics." In: *Proceedings of the National Institute of Sciences (Calcutta)* 2 (1936), pp. 49–55.

[117] Geoffrey J. McLachlan. "Mahalanobis distance." In: *Resonance* 4.6 (1999), pp. 20–26. ISSN: 0973-712X. DOI: 10.1007/BF02834632.

[118] Roy De Maesschalck, Delphine Jouan-Rimbaud, and Desire L. Massart. "The Mahalanobis distance." In: *Chemometrics and Intelligent Laboratory Systems* 50.1 (2000), pp. 1–18. ISSN: 0169-7439. DOI: 10.1016/S0169-7439(99)00047-7.

[119] Paul Fieguth. *An Introduction to Pattern Recognition and Machine Learning*. Cham, Switzerland: Springer, 2022. ISBN: 978-3-030-95995-1. DOI: 10.1007/978-3-030-95995-1.

[120] Svante Wold, Michael Sjöström, and Lennart Eriksson. "PLS-regression: a basic tool of chemometrics." In: *Chemometrics and Intelligent Laboratory Systems* 58.2 (2001), pp. 109–130. ISSN: 0169-7439. DOI: 10.1016/S0169-7439(01)00155-1.

[121] Svante Wold et al. "Multivariate Data Analysis in Chemistry." In: *Chemometrics: Mathematics and Statistics in Chemistry*. Ed. by Bruce R. Kowalski. Dordrecht, The Netherlands: Springer, 1984, pp. 17–95. ISBN: 978-94-017-1026-8. DOI: 10.1007/978-94-017-1026-8_2.

[122] Paul Geladi and Bruce R. Kowalski. "Partial least-squares regression: a tutorial." In: *Analytica Chimica Acta* 185 (1986), pp. 1–17. ISSN: 0003-2670. DOI: 10.1016/0003-2670(86)80028-9.

[123] Sijmen De Jong. "SIMPLS: An alternative approach to partial least squares regression." In: *Chemometrics and Intelligent Laboratory Systems* 18.3 (1993), pp. 251–263. ISSN: 0169-7439. DOI: 10.1016/0169-7439(93)85002-X.

[124] Avraham Lorber and Bruce R. Kowalski. "A Note on the Use of the Partial Least-Squares Method for Multivariate Calibration." In: *Applied Spectroscopy* 42.8 (1988), pp. 1572–1574. DOI: 10.1366/0003702884429481.

[125] Jan Larsen and Cyril Goutte. "On optimal data split for generalization estimation and model selection." In: *Neural Networks for Signal Processing IX: Proceedings of the 1999 IEEE Signal Processing Society Workshop (Cat. No.98TH8468)* (Madison, WI, USA, Aug. 25, 1999). 1999, pp. 225–234. DOI: 10.1109/NNSP.1999.788141.

[126] Kevin K. Dobbin and Richard M. Simon. "Optimally splitting cases for training and testing high dimensional classifiers." In: *BMC Medical Genomics* 4.1 (2011), p. 31. ISSN: 1755-8794. DOI: 10.1186/1755-8794-4-31.

[127] Quang Hung Nguyen et al. "Influence of Data Splitting on Performance of Machine Learning Models in Prediction of Shear Strength of Soil." In: *Mathematical Problems in Engineering* (2021). Ed. by Yu-Sheng Shen, p. 4832864. ISSN: 1024-123X. DOI: 10.1155/2021/4832864.

[128] Joseph Roshan Vengazhiyil. "Optimal ratio for data splitting." In: *Statistical Analysis and Data Mining: The ASA Data Science Journal* 15.4 (2022), pp. 531–538. DOI: 10.1002/sam.11583.

[129] Robert Sanders. "The Pareto Principle: its Use and Abuse." In: *Journal of Services Marketing* 1.2 (Jan. 1987), pp. 37–40. ISSN: 0887-6045. DOI: 10.1108/eb024706.

[130] Stuart Geman, Elie Bienenstock, and René Doursat. "Neural Networks and the Bias/Variance Dilemma." In: *Neural Computation* 4.1 (1992), pp. 1–58. DOI: 10.1162/neco.1992.4.1.1.

[131] Mikhail Belkin et al. "Reconciling modern machine-learning practice and the classical bias–variance trade-off." In: *Proceedings of the National Academy of Sciences* 116.32 (2019), pp. 15849–15854. DOI: 10.1073/pnas.1903070116.

[132] Scott Fortmann-Roe. *Understanding the Bias-Variance Tradeoff*. June 2012. URL: http://scott.fortmann-roe.com/docs/BiasVariance.html. [Online; accessed March 26, 2023].

[133] Sylvain Arlot and Alain Celisse. "A survey of cross-validation procedures for model selection." In: *Statistics Surveys* 4 (2010), pp. 40–79. ISSN: 1935-7516. DOI: 10.1214/09-SS054.

[134] CERN Data Centre & Invenio. *Zenodo*. URL: https://zenodo.org/. [Online; accessed December 1, 2022].

[135] James P. Fox, Jadd R. Brammall, and Prasad K. D. V. Yarlagadda. "Determination of the financial impact of machine downtime on the post large letters sorting process." In: *Journal of Achievements in Materials and Manufacturing Engineering* 31.2 (2008), pp. 732–738.

[136] B. K. N. Rao. "Condition monitoring and the integrity of industrial systems." In: *Handbook of Condition Monitoring: Techniques and Methodology*. Ed. by Alan Davies. Dordrecht, The Netherlands: Springer, 1998. Chap. 1, pp. 3–34. ISBN: 978-94-011-4924-2. DOI: 10.1007/978-94-011-4924-2_1.

[137] Hans J. Matthies and Karl T. Renius. *Einführung in die Ölhydraulik*. Wiesbaden, Germany: Springer Vieweg, 2021. ISBN: 978-3-658-35672-9. DOI: 10.1007/978-3-658-35673-6.

[138] Tizian Schneider, Steffen Klein, and Manuel Bastuck. *Condition monitoring of hydraulic systems Data Set at ZeMA*. Apr. 2018. DOI: 10.5281/zenodo.1323611. [Data set].

[139] Nikolai Helwig. "Zustandsbewertung industrieller Prozesse mittels multivariater Sensordatenanalyse am Beispiel hydraulischer und elektromechanischer Antriebssysteme." PhD thesis. Dept. Systems Engineering, Saarland University, Saarbruecken, 2018. DOI: 10.22028/D291-27896.

[140] 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)* (Pisa, Italy, May 11–14, 2015). 2015, pp. 210–215. DOI: 10.1109/I2MTC.2015.7151267.

[141] Steffen Klein. "Physikalisch motivierte Extraktion von Merkmalen zur multivariaten Schadens- und Verschleißdetektion elektromechanischer Zylinder." Master thesis. Dept. Systems Engineering, Saarland University, Saarbruecken, 2018.

[142] Tanja Dorst. *Sensor data set of 3 electromechanical cylinder at ZeMA testbed (2kHz)*. May 2019. DOI: 10.5281/zenodo.3929385. [Data set].

[143] Festo SE & Co. KG. *Electric cylinders ESBF, with spindle drive*. URL: https://www.festo.com/cat/en-gb_gb/data/doc_ENUS/PDF/US/ESBF_ENUS.PDF. [Online; accessed December 2, 2022].

[144] Heinrich Dubbel. *Dubbel Taschenbuch für den Maschinenbau 2: Anwendungen*. Ed. by Beate Bender and Dietmar Göhlich. 26th ed. Berlin + Heidelberg, Germany: Springer Vieweg, 2020. ISBN: 978-3-662-59712-5.

[145] Jochen Forstmann. "Kugelgewindetriebe im Einsatz an Kunststoffspritzgießmaschinen – Lebensdauerprognose und Optimierung." PhD thesis. Faculty of Engineering, University of Duisburg-Essen, Duisburg + Essen, 2010. URL: https://duepublico2.uni-due.de/receive/duepublico_mods_00022163.

[146] Tanja Dorst et al. "Uncertainty-aware data pipeline of calibrated MEMS sensors used for machine learning." In: *Measurement: Sensors* 22 (2022), p. 100376. ISSN: 2665-9174. DOI: 10.1016/j.measen.2022.100376.

[147] Peggy L. Jenkins et al. "Activity patterns of Californians: Use of and proximity to indoor pollutant sources." In: *Atmospheric Environment. Part A. General Topics* 26.12 (1992), pp. 2141–2148. ISSN: 0960-1686. DOI: 10.1016/0960-1686(92)90402-7.

[148] Neil E. Kleoeis et al. "The National Human Activity Pattern Survey (NHAPS): a resource for assessing exposure to environmental pollutants." In: *Journal of Exposure Science & Environmental Epidemiology* 11.3 (2001), pp. 231–252. ISSN: 1559-064X. DOI: 10.1038/sj.jea.7500165.

[149] Sabine Brasche and Wolfgang Bischof. "Daily time spent indoors in German homes – Baseline data for the assessment of indoor exposure of German occupants." In: *International Journal of Hygiene and Environmental Health* 208.4 (2005), pp. 247–253. ISSN: 1438-4639. DOI: 10.1016/j.ijheh.2005.03.003.

[150] Jan Sundell. "On the history of indoor air quality and health." In: *Indoor air* 14 (Suppl. 7) (2004), pp. 51–58.

[151] Arja Asikainen et al. "Reducing burden of disease from residential indoor air exposures in Europe (HEALTHVENT project)." In: *Environmental Health* 15.1 (2016), S35. ISSN: 1476-069X. DOI: 10.1186/s12940-016-0101-8.

[152] Carrie A. Redlich, Judy Sparer, and Mark R. Cullen. "Sick-building syndrome." In: *The Lancet* 349.9057 (1997), pp. 1013–1016. ISSN: 0140-6736. DOI: 10.1016/S0140-6736(96)07220-0.

[153] Joann Ten Brinke et al. "Development of New Volatile Organic Compound (VOC) Exposure Metrics and their Relationship to "Sick Building Syndrome" Symptoms." In: *Indoor Air* 8.3 (1998), pp. 140–152. DOI: 10.1111/j.1600-0668.1998.t01-1-00002.x.

[154] Andrew P. Jones. "Indoor air quality and health." In: *Atmospheric Environment* 33.28 (1999), pp. 4535–4564. ISSN: 1352-2310. DOI: 10.1016/S1352-2310(99)00272-1.

[155] Wen-Tien Tsai. "An overview of health hazards of volatile organic compounds regulated as indoor air pollutants." In: *Reviews on Environmental Health* 34.1 (2019), pp. 81–89. DOI: 10.1515/reveh-2018-0046.

[156] Jill D. Fenske and Suzanne E. Paulson. "Human breath emissions of VOCs." In: *Journal of the Air & Waste Management Association (1995)* 49.5 (May 1999), pp. 594–598. ISSN: 1096-2247. DOI: 10.1080/10473289.1999.10463831.

[157] Nijing Wang et al. "Emission Rates of Volatile Organic Compounds from Humans." In: *Environmental Science & Technology* 56.8 (Apr. 2022), pp. 4838–4848. ISSN: 0013-936X. DOI: 10.1021/acs.est.1c08764.

[158] Max von Pettenkofer. *Über den Luftwechsel in Wohngebäuden.* München, Germany: Cotta, 1858. URL: https://opacplus.bsb-muenchen.de/title/BV013009721.

[159] Stephen K. Brown et al. "Concentrations of Volatile Organic Compounds in Indoor Air – A Review." In: *Indoor Air* 4.2 (1994), pp. 123–134. DOI: 10.1111/j.1600-0668.1994.t01-2-00007.x.

[160] Wil A. Spaul. "Building-related factors to consider in indoor air quality evaluations." In: *The Journal of allergy and clinical immunology* 94.2, Part 2 (Aug. 1994), pp. 385–389. ISSN: 0091-6749 (Print).

[161] Ananya Dey. "Semiconductor metal oxide gas sensors: A review." In: *Materials Science and Engineering: B* 229 (2018), pp. 206–217. ISSN: 0921-5107. DOI: 10.1016/j.mseb.2017.12.036.

[162] Andreas Schütze and Tilman Sauerwald. "Dynamic operation of semiconductor sensors." In: *Semiconductor Gas Sensors*. Ed. by Raivo Jaaniso and Ooi Kiang Tan. 2nd. Woodhead Publishing Series in Electronic and Optical Materials. Woodhead Publishing, 2020, pp. 385–412. ISBN: 978-0-08-102559-8. DOI: 10.1016/B978-0-08-102559-8.00012-4.

[163] Hartmut Eicker. "Method and apparatus for determining the concentration of one gaseous component in a mixture of gases." Pat. US4012692A. Mar. 1977. URL: http://www.google.tl/patents/US4012692.

[164] Yo Kato, Kenichi Yoshikawa, and Maki Kitora. "Temperature-dependent dynamic response enables the qualification and quantification of gases by a single sensor." In: *Sensors and Actuators B: Chemical* 40.1 (1997), pp. 33–37. ISSN: 0925-4005. DOI: 10.1016/S0925-4005(97)80196-7.

[165] Andrew P. Lee and Brian J. Reedy. "Temperature modulation in semiconductor gas sensing." In: *Sensors and Actuators B: Chemical* 60.1 (1999), pp. 35–42. ISSN: 0925-4005. DOI: 10.1016/S0925-4005(99)00241-5.

[166] Tobias Baur, Andreas Schütze, and Tilman Sauerwald. "Optimierung des temperaturzyklischen Betriebs von Halbleitergassensoren." In: *tm - Technisches Messen* 82.4 (2015), pp. 187–195. DOI: 10.1515/teme-2014-0007.

[167] Tobias Baur et al. "Field Study of Metal Oxide Semiconductor Gas Sensors in Temperature Cycled Operation for Selective VOC Monitoring in Indoor Air." In: *Atmosphere* 12.5 (2021). ISSN: 2073-4433. DOI: 10.3390/atmos12050647.

[168] Johannes Amann, Tobias Baur, and Caroline Schultealbert. *Measuring Hydrogen in Indoor Air with a Selective Metal Oxide Semiconductor Sensor: Dataset*. Mar. 2021. DOI: 10.5281/zenodo.4593853. [Data set].

[169] Caroline Schultealbert et al. “Measuring Hydrogen in Indoor Air with a Selective Metal Oxide Semiconductor Sensor.” In: *Atmosphere* 12.3 (2021). ISSN: 2073-4433. DOI: 10.3390/atmos12030366.

[170] Sensirion AG. *Datasheet SGP30*. May 2020. URL: https://sensirion.com/media/documents/984E0DD5/61644B8B/Sensirion_Gas_Sensors_Datasheet_SGP30.pdf. [Online; accessed November 26, 2022].

[171] Daniel Rüffer, Felix Hoehne, and Johannes Bühler. “New Digital Metal-Oxide (MOx) Sensor Platform.” In: *Sensors* 18.4 (2018). ISSN: 1424-8220. DOI: 10.3390/s18041052.

[172] Michael Hauptmann et al. “Mortality from solid cancers among workers in formaldehyde industries.” In: *American journal of epidemiology* 159.12 (June 2004), pp. 1117–1130. ISSN: 0002-9262. DOI: 10.1093/aje/kwh174.

[173] Luoping Zhang. *Formaldehyde*. Issues in Toxicology. The Royal Society of Chemistry, 2018. ISBN: 978-1-78262-973-3. DOI: 10.1039/9781788010269.

[174] NTP (National Toxicology Program). *Report on Carcinogens, Fifteenth Edition*. 2021. DOI: 10.22427/NTP-OTHER-1003. [Online; accessed November 27, 2022].

[175] Kimmo Koistinen et al. “The INDEX project: executive summary of a European Union project on indoor air pollutants.” In: *Allergy* 63.7 (2008), pp. 810–819. DOI: 10.1111/j.1398-9995.2008.01740.x.

[176] Beat Meyer and Karl Hermanns. “Formaldehyde Release from Pressed Wood Products.” In: *Formaldehyde*. Ed. by Victor Turoski. Vol. 210. Advances in Chemistry. American Chemical Society, Aug. 1985, pp. 101–116. ISBN: 978-0-841-20903-9. DOI: 10.1021/ba-1985-0210.ch008.

[177] James M. Donald, Kim Hooper, and Claudia Hopenhayn-Rich. “Reproductive and developmental toxicity of toluene: a review.” In: *Environmental Health Perspectives* 94 (Aug. 1991), pp. 237–244. DOI: 10.1289/ehp.94-1567945.

[178] Tobias Baur et al. “Random gas mixtures for efficient gas sensor calibration.” In: *Journal of Sensors and Sensor Systems* 9.2 (2020), pp. 411–424. DOI: 10.5194/jsss-9-411-2020.

[179] Michael D. McKay, Richard J. Beckman, and William J. Conover. “A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code.” In: *Technometrics* 21.2 (1979), pp. 239–245. DOI: 10.2307/1268522.

[180] Wei-Liem Loh. "On Latin hypercube sampling." In: *The Annals of Statistics* 24.5 (1996), pp. 2058–2080. DOI: 10.1214/aos/1069362310.

[181] Johannes Amann et al. "Bewertung der Innenraumluftqualität über VOC-Messungen mit Halbleitergassensoren - Kalibrierung, Feldtest, Validierung." In: *tm - Technisches Messen* 88.S1 (2021), S89–S94. DOI: 10.1515/teme-2021-0058.

[182] Graham Pryor, ed. *Managing Research Data*. Facet Publishing, 2012. ISBN: 978-1-85604-891-0. DOI: 10.29085/9781856048910.

[183] Neela Enke et al. "The user's view on biodiversity data sharing — Investigating facts of acceptance and requirements to realize a sustainable use of research data —." In: *Ecological Informatics* 11 (2012), pp. 25–33. ISSN: 1574-9541. DOI: 10.1016/j.ecoinf.2012.03.004.

[184] Philippe Grandcolas. "The Rise of "Digital Biology": We need not only open, FAIR but also sustainable data!" In: *Biodiversity Information Science and Standards* (Leiden, The Netherlands, Oct. 22–25, 2019). Vol. 3. 2019, e37508. DOI: 10.3897/biss.3.37508.

[185] Mark D. Wilkinson et al. "The FAIR Guiding Principles for scientific data management and stewardship." In: *Scientific Data* 3.1 (2016), p. 160018. ISSN: 2052-4463. DOI: 10.1038/sdata.2016.18.

[186] Natasha Simons et al. *The State of Open Data 2021*. Digital Science, Nov. 2021. DOI: 10.6084/m9.figshare.17061347.v1.

[187] Shelley Stall et al. "Make scientific data FAIR." In: *Nature* 570 (2019), pp. 27–29. DOI: 10.1038/d41586-019-01720-7.

[188] DCMI Usage Board. *DCMI Metadata Terms*. DCMI Recommendation. Dublin Core Metadata Initiative, 2020. URL: http://dublincore.org/specifications/dublin-core/dcmi-terms/2020-01-20/.

[189] Daniel Hutzschenreuter et al. *SmartCom Digital System of Units (D-SI)*. Tech. rep. July 2020. DOI: 10.5281/zenodo.3816686.

[190] W3C. *RDF Schema 1.1*. Tech. rep. 2014. URL: https://www.w3.org/TR/rdf-schema/.

[191] Krzysztof Janowicz et al. "SOSA: A lightweight ontology for sensors, observations, samples, and actuators." In: *Journal of Web Semantics* 56 (2019), pp. 1–10. ISSN: 1570-8268. DOI: 10.1016/j.websem.2018.06.003.

[192] Christophe Bahim et al. "The FAIR Data Maturity Model: An Approach to Harmonise FAIR Assessments." In: *Data Science Journal* 19.1 (2020), pp. 1–7. DOI: 10.5334/dsj-2020-041.

[193] FAIR Data Maturity Model Working Group. *FAIR Data Maturity Model. Specification and Guidelines*. June 2020. DOI: 10.15497/rda00050.

[194] Ron Davies. *Industry 4.0: Digitalisation for productivity and growth*. Tech. rep. 2015. URL: https://policycommons.net/artifacts/1335939/industry-40/.

[195] Gil Press. "Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Survey Says." In: *Forbes* (Mar. 2016). URL: https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/?sh=60afcc3f6f63. [Online; accessed March 1, 2023].

[196] Hui Yie Teh, Andreas W. Kempa-Liehr, and Kevin I-Kai Wang. "Sensor data quality: a systematic review." In: *Journal of Big Data* 7.1 (2020), pp. 1–49. ISSN: 2196-1115. DOI: 10.1186/s40537-020-0285-1.

[197] Venkat N. Gudivada, Amy Apon, and Junhua Ding. "Data Quality Considerations for Big Data and Machine Learning: Going Beyond Data Cleaning and Transformations." In: *International Journal on Advances in Software* 10.1 & 2 (2017), pp. 1–20. ISSN: 1942-2628. URL: http://www.iariajournals.org/software/soft_v10_n12_2017_paged.pdf.

[198] Abhinav Jain et al. "Overview and Importance of Data Quality for Machine Learning Tasks." In: *KDD'20: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Online, July 6–10, 2020). Association for Computing Machinery, 2020, pp. 3561–3562. ISBN: 978-1-4503-7998-4. DOI: 10.1145/3394486.3406477.

[199] STMicroelectronics. *STM32F767ZI*. URL: https://www.st.com/en/microcontrollers-microprocessors/stm32f767zi.html. [Online; accessed December 5, 2022].

[200] Sascha Eichstädt et al. "Toward Smart Traceability for Digital Sensors and the Industrial Internet of Things." In: *Sensors* 21.6 (2021). ISSN: 1424-8220. DOI: 10.3390/s21062019.

[201] Bosch Sensortec GmbH. *Data sheet BMA280 - Digital, triaxial acceleration sensor*. Oct. 2021. URL: https://edn.com/wp-content/uploads/bst-bma280-ds000.pdf. [Online; accessed February 26, 2023].

[202] InvenSense Inc. *MPU-9250 Product Specification*. June 2016. URL: https://invensense.tdk.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf. [Online; accessed February 26, 2023].

[203] TE Connectivity. *MS5837-02BA*. Dec. 2019. URL: https://www.te.com/commerce/DocumentDelivery/DDEController?Action=showdoc&DocId=Data+Sheet%7FMS5837-02BA01%7FA8%7Fpdf%7FEnglish%7FENG_DS_MS5837-02BA01_A8.pdf%7FCAT-BLPS0059. [Online; accessed February 26, 2023].

[204] Tanja Dorst, Sascha Eichstädt, and Andreas Schütze. *Report on the implementation of sensor calibration and sensor network methods for metrological infrastructure improvement of the ZEMA testbed, including provision of test data for ML development*. Deliverable D7 of the EMPIR Project 17IND12 Met4FoF. 2021.

[205] Benedikt Seeger. *Software for the Met4FoF Smartup Unit*. Version V3. Dec. 2022. URL: https://github.com/Met4FoF/Met4FoF-SmartUpUnit.

[206] Andrew F. Siegel. "Robust regression using repeated medians." In: *Biometrika* 69.1 (1982), pp. 242–244. ISSN: 0006-3444. DOI: 10.1093/biomet/69.1.242.

[207] Sascha Eichstädt et al. *PyDynamic*. URL: https://pypi.org/project/PyDynamic/. [Online; accessed December 6, 2022].

[208] Björn Ludwig et al. *PTB-M4D/PyDynamic: v2.3.1*. Version 2.3.1. Nov. 2022. DOI: 10.5281/zenodo.1489877.

[209] David R. White and Peter Saunders. "The propagation of uncertainty with calibration equations." In: 18.7 (June 2007), pp. 2157–2169. DOI: 10.1088/0957-0233/18/7/047.

[210] David R. White. "Propagation of Uncertainty and Comparison of Interpolation Schemes." In: *International Journal of Thermophysics* 38.3 (2017), p. 39. ISSN: 1572-9567. DOI: 10.1007/s10765-016-2174-6.

[211] S. Eichstädt et al. "Deconvolution filters for the analysis of dynamic measurement processes: a tutorial." In: *Metrologia* 47.5 (Aug. 2010), pp. 522–533. DOI: 10.1088/0026-1394/47/5/003.

[212] Alfred Link and Clemens Elster. "Uncertainty evaluation for IIR (Infinite Impulse Response) Filtering using a State-Space Approach." In: *Measurement Science and Technology* 20.5 (2009). DOI: 10.1088/0957-0233/20/5/055104.

[213] Tanja Dorst et al. "Providing FAIR and metrologically traceable data sets - a case study." In: *IMEKO TC6 International Conference on Metrology and Digital Transformation* (Berlin, Germany, Sept. 19–21, 2022). 2022. URL: https://www.m4dconf2022.ptb.de/fileadmin/documents/m4dconf2022/Material/Paper/IMEKOTC6-M4Dconf-2022-P13-DORST-et-al.pdf.

[214] Tanja Dorst et al. "A Case Study on providing FAIR and metrologically traceable data sets." In: *Acta IMEKO* 12.1, article 5 (Mar. 2023), pp. 1–6. ISSN: 2221-870X.

[215] Tanja Dorst, Maximilian Gruber, and Anupam Prasad Vedurmudi. *Sensor data set of one electromechanical cylinder at ZeMA testbed (ZeMA DAQ and Smart-Up Unit)*. Sept. 2021. DOI: 10.5281/zenodo.5185953. [Data set].

[216] Thyago P. Carvalho et al. "A systematic literature review of machine learning methods applied to predictive maintenance." In: *Computers & Industrial Engineering* 137 (2019), p. 106024. ISSN: 0360-8352. DOI: 10.1016/j.cie.2019.106024.

[217] Juan Pablo Usuga Cadavid et al. "Machine learning applied in production planning and control: a state-of-the-art in the era of industry 4.0." In: *Journal of Intelligent Manufacturing* 31.6 (2020), pp. 1531–1558. ISSN: 1572-8145. DOI: 10.1007/s10845-019-01531-7.

[218] Ying Lei et al. "Algorithms for time synchronization of wireless structural monitoring sensors." In: *Earthquake Engineering & Structural Dynamics* 34.6 (2005), pp. 555–573. DOI: 10.1002/eqe.432.

[219] Tanja Dorst et al. "Influence of synchronization within a sensor network on machine learning results." In: *Journal of Sensors and Sensor Systems* 10.2 (2021), pp. 233–245. DOI: 10.5194/jsss-10-233-2021.

[220] Henry J. Landau. "Sampling, data transmission, and the Nyquist rate." In: *Proceedings of the IEEE* 55.10 (1967), pp. 1701–1706. DOI: 10.1109/PROC.1967.5962.

[221] Tizian Schneider et al. "Big Data Analytik mit automatisierter Signalverarbeitung für Condition Monitoring." In: *Sensoren und Messsysteme, 19. ITG/GMA-Fachtagung* (Nürnberg, Germany, June 26–27, 2018). Ed. by Leonhard M. Reindl and Jürgen Wöllenstein. VDE-Verlag, 2018, pp. 259–262. ISBN: 978-3-8007-4866-2.

[222] Tanja Dorst et al. “Uncertainty-aware automated machine learning toolbox.” In: *tm - Technisches Messen* 90.3 (2023), pp. 141–153. DOI: 10.1515/teme-2022-0042.

[223] Sascha Eichstädt et al. “Efficient implementation of a Monte Carlo method for uncertainty evaluation in dynamic measurements.” In: *Metrologia* 49.3 (Apr. 2012), pp. 401–410. DOI: 10.1088/0026-1394/49/3/401.

[224] Tanja Dorst et al. “Propagation of uncertainty for an Adaptive Linear Approximation algorithm.” In: *SMSI 2020 - Sensor and Measurement Science International Proceedings*. 2020, pp. 366–367. DOI: 10.5162/SMSI2020/E2.3.

[225] Tanja Dorst et al. “GUM2ALA – Uncertainty propagation algorithm for the Adaptive Linear Approximation according to the GUM.” In: *SMSI 2021 - Sensor and Measurement Science International Proceedings* (Online, May 3–6, 2021). Digital Conference, 2021, pp. 314–315. ISBN: 978-3-9819376-4-0. DOI: 10.5162/SMSI2021/D1.1.

[226] Maximilian Gruber et al. “Discrete wavelet transform on uncertain data: Efficient online implementation for practical applications.” In: *Series on Advances in Mathematics for Applied Sciences*. Ed. by Franco Pavese et al. WORLD SCIENTIFIC, Jan. 2022, pp. 249–261. DOI: 10.1142/9789811242380_0014.

[227] Harold S. M. Coxeter. *Regular Polytopes*. 3rd ed. Dover Publications Inc., 1974. ISBN: 978-0-486-61480-9.

[228] Michel C. Jeruchim, Philip Balaban, and K. Sam Shanmugan. *Simulation of Communication Systems*. 2nd ed. New York, NY, USA: Springer, 2000. ISBN: 978-0-306-46971-8. DOI: 10.1007/b117713.

[229] Vasilis Z. Marmarelis. *Nonlinear Dynamic Modeling of Physiological Systems*. John Wiley & Sons, Inc., 2004. ISBN: 978-0-471-46960-5. DOI: 10.1002/9780471679370.

[230] Xinhua Zhang. “Gaussian Distribution.” In: *Encyclopedia of Machine Learning*. Ed. by Claude Sammut and Geoffrey I. Webb. Boston, MA, USA: Springer, 2010, pp. 425–428. ISBN: 978-0-387-30164-8. DOI: 10.1007/978-0-387-30164-8_323.

[231] Yadolah Dodge. “Uniform Distribution.” In: *The Concise Encyclopedia of Statistics*. New York, NY: Springer New York, 2008, pp. 167–168. ISBN: 978-0-387-32833-1. DOI: 10.1007/978-0-387-32833-1_411.

[232] William R. Bennett. “Spectra of quantized signals.” In: *The Bell System Technical Journal* 27.3 (1948), pp. 446–472. DOI: 10.1002/j.1538-7305.1948.tb01340.x.

[233] Yannick Robin et al. “High-Performance VOC Quantification for IAQ Monitoring Using Advanced Sensor Systems and Deep Learning.” In: *Atmosphere* 12.11 (2021). ISSN: 2073-4433. DOI: 10.3390/atmos12111487.

[234] Sebastien C. Wong et al. “Understanding Data Augmentation for Classification: When to Warp?” In: *Proceedings of International Conference on Digital Image Computing: Techniques and Applications (DICTA)* (Gold Coast, Australia, Nov. 30–Dec. 2, 2016). 2016, pp. 1–6. DOI: 10.1109/DICTA.2016.7797091.

[235] Connor Shorten and Taghi M. Khoshgoftaar. “A survey on Image Data Augmentation for Deep Learning.” In: *Journal of Big Data* 6.1 (2019), p. 60. ISSN: 2196-1115. DOI: 10.1186/s40537-019-0197-0.

[236] Marko Robnik-Šikonja and Igor Kononenko. “An adaptation of Relief for attribute estimation in regression.” In: *Proceedings of the Fourteenth International Conference on Machine Learning (ICML)* (Nashville, TN, USA, July 8–12, 1997). Morgan Kaufmann Publishers Inc., 1997, pp. 296–304.

[237] Johannes F. Amann. “Möglichkeiten und Grenzen des Einsatzes von Halbleitergassensoren im temperaturzyklischen Betrieb für die Messung der Innenraumluftqualität – Kalibrierung, Feldtest, Validierung.” Master thesis. Dept. Systems Engineering, Saarland University, Saarbruecken, 2021.

Danksagung

Die vorliegende Dissertation ist im Rahmen einer Kooperation des Zentrums für Mechatronik und Automatisierungstechnik (ZeMA) gGmbH mit der Physikalisch-Technischen Bundesanstalt (PTB) entstanden.

An dieser Stelle möchte ich allen beteiligten Personen meinen großen Dank aussprechen, die mich bei der Anfertigung meiner Dissertation unterstützt haben.

An erster Stelle danke ich daher Prof. Dr. Andreas Schütze als Doktorvater und Dr. Sascha Eichstädt, ohne die diese Promotion nicht möglich gewesen wäre. Beide hatten in den vergangenen vier Jahren immer ein offenes Ohr und haben mit intensiven Diskussionsrunden maßgeblich zum Gelingen dieser Arbeit beigetragen. Auch möchte ich mich für ausgezeichnete Betreuung und die Unterstützung bei der Durchführung der gesamten Arbeit bei beiden bedanken.

Darüber hinaus gilt mein Dank Prof. Rainer Tutsch, der sich bereit erklärt hat, das Zweitgutachten zu übernehmen.

Mindestens genauso wichtig war die angenehme und kollegiale Arbeitsatmosphäre am Lehrstuhl für Messtechnik (LMT) der Universität des Saarlandes, am ZeMA und bei der PTB. Daher danke ich meinen Kolleginnen und Kollegen bei der PTB, die trotz der sehr wenigen persönlichen Treffen mir immer das Gefühl gaben, zum Team zu gehören. Insbesondere möchte ich an dieser Stelle Maximilian Gruber erwähnen, mit dem ich in der Zeit meiner Promotion immer wieder gute Diskussionen per Videokonferenz geführt habe. Auch danke ich meinen Kollegen der DESS-Gruppe – Christian, Christopher, Eric, Payman, Sebastian, Steffen, Tizian, Yannick – für die erfolgreiche Zusammenarbeit und schöne gemeinsame Zeit am ZeMA. Leider habe ich in der ganzen Zeit nicht viele persönliche Kontakte zu meinen Kolleginnen und Kollegen der Gasmesstechnik gehabt. Für die wenigen, aber guten Diskussionen und Anregungen möchte ich mich dennoch bei Christian, Henrik, Johannes, Julian, My Sa und Oliver herzlich bedanken. Auch möchte ich Christiana und Harald an dieser Stelle meinen Dank aussprechen, die stets ein offenes Ohr hatten und bei organisatorischen Angelegenheiten tatkräftig unterstützt haben.

Außerdem danke ich zwei ehemaligen Mitarbeitern des LMT: Eliseo Pignanelli und Marco Schott. Beide haben mich auf meinem Weg mit Rat und produktiven Gesprächen begleitet. Besonders danken möchte ich an dieser Stelle auch Dr. Karsten Kühn, welcher mich überhaupt erst auf die Idee einer Promotion gebracht und den Kontakt zum LMT hergestellt hatte.

Mein tiefer Dank geht an meine Eltern, die mein Studium ermöglicht haben und mir stets moralischen Rückhalt geben. Auch danke ich meinen Freunden für ihre Geduld, Ermutigungen und guten Zusprüche in den letzten Jahren. Des Weiteren sage ich zudem Christopher Benkert, Christopher Schnur und meiner Schwester Katja Dorst Danke, die diese Arbeit aufmerksam auf Rechtschreib- und Grammatikfehler überprüft haben und auch Yannick Robin, für die inhaltliche Korrekturlesung. Besonders danke ich an dieser Stelle Isabelle, die auch in schwierigen Zeiten immer da ist und natürlich Mia, die mich immer wieder zum Lachen brachte.

Abschließend danke ich meinem Mann Christian von ganzem Herzen. Mit seiner endlosen Geduld und seiner andauernden Motivation hat er mir das Leben in den Stressphasen einfacher gemacht. Er hat nie aufgehört an mich zu glauben, auch wenn ich mal wieder den Mut verloren hatte.

Danke!