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],

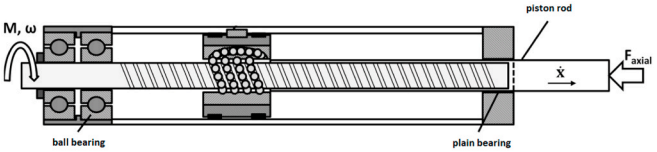

Fig. 2. Simplified structure of an EMC with a spindle drive (adapted from Ref. [5]).

- 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

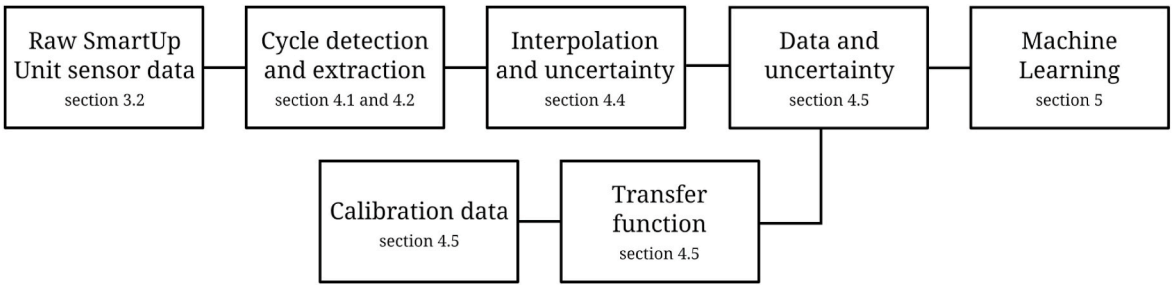

Fig. 1. Overview of the data processing pipeline.

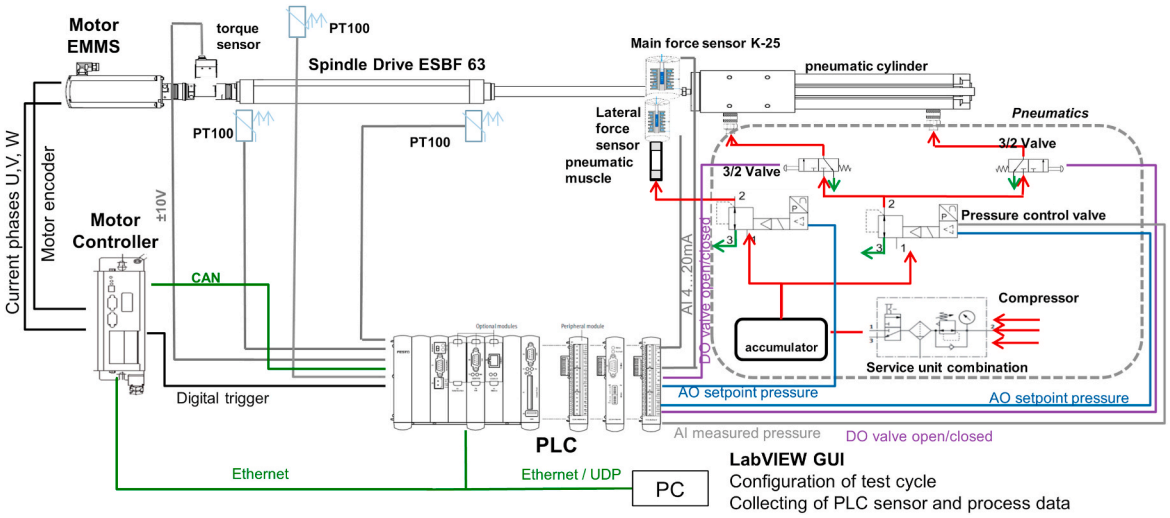

Fig. 3. Scheme of the test bed [6].

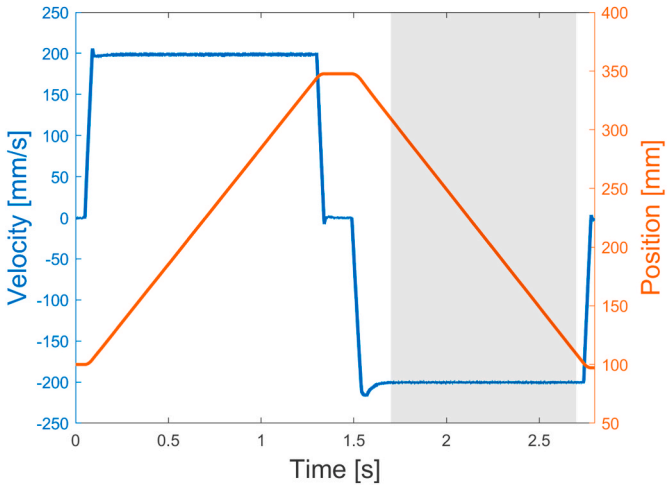

Fig. 4. Working cycle of the EMC test bed.

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

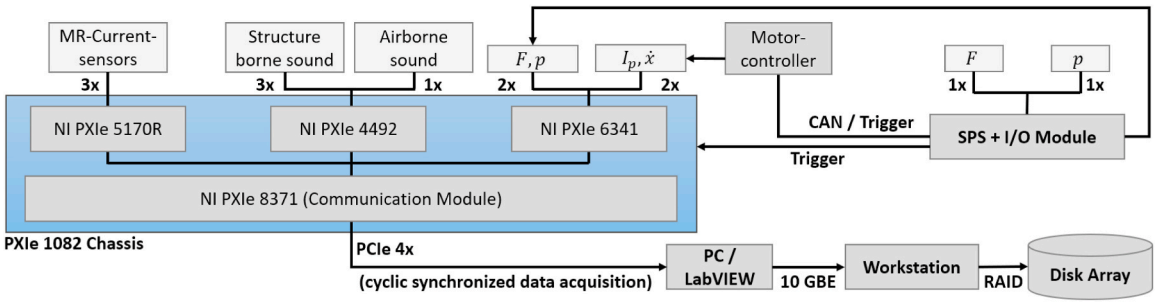

Fig. 5. Data acquisition unit of the EMC test bed (adapted from Ref. [5]).

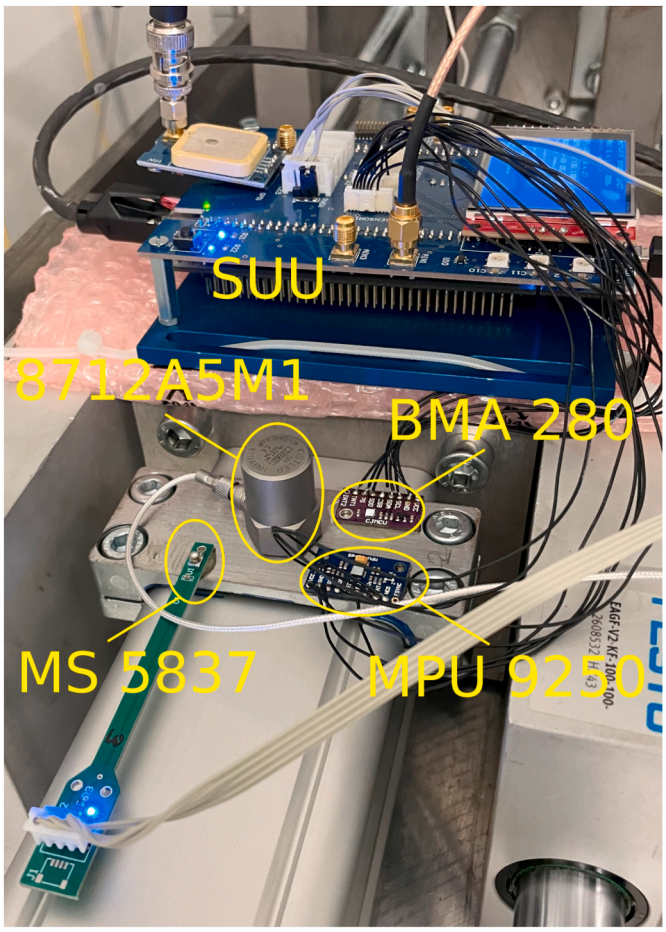

Fig. 6. Installation of the SUU and its three sensors in the EMC test bed. The Kistler 8712A5M1 acceleration sensor of the ZeMA DAQ is also installed at the same location.

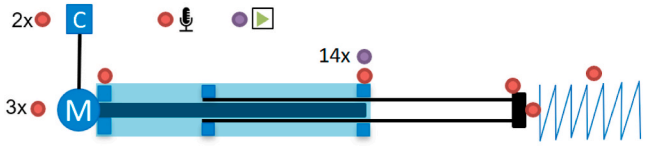

Fig. 7. SUU sensors (purple) and ZeMA DAQ sensors (red) localization with respect to the EMC. The green triangle symbolizes the trigger, which indicates the start of a cycle and is recorded by both the ZeMA DAQ and the SUU. (For interpretation of the references to colour in this figure legend, the reader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

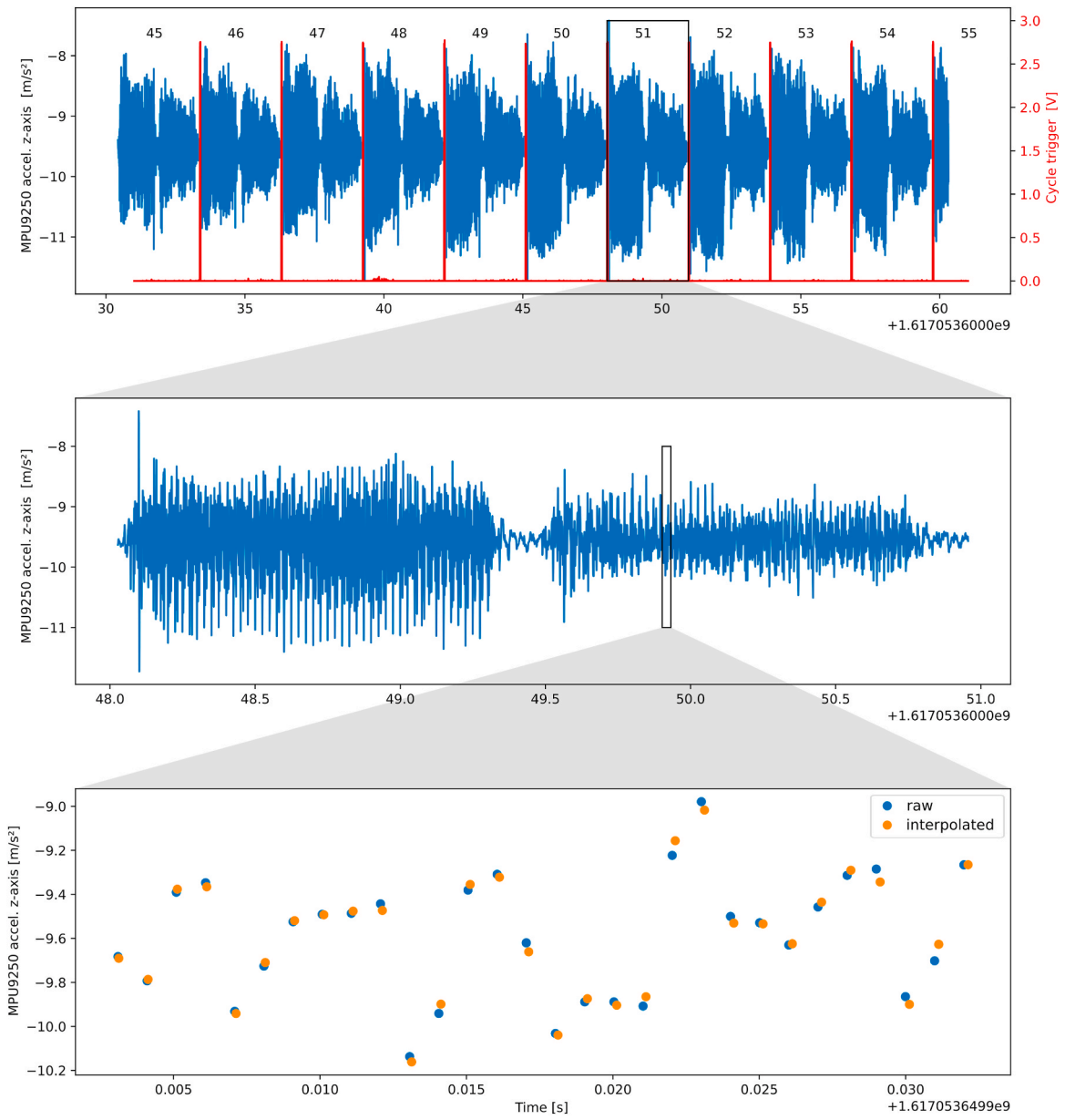

Fig. 8. Breakdown of preprocessing steps. Top: Extract cycle count from trigger signal. Mid: extracted 51 cycle. Bottom: detail of interpolation at given sample rate.

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].

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

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

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

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

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

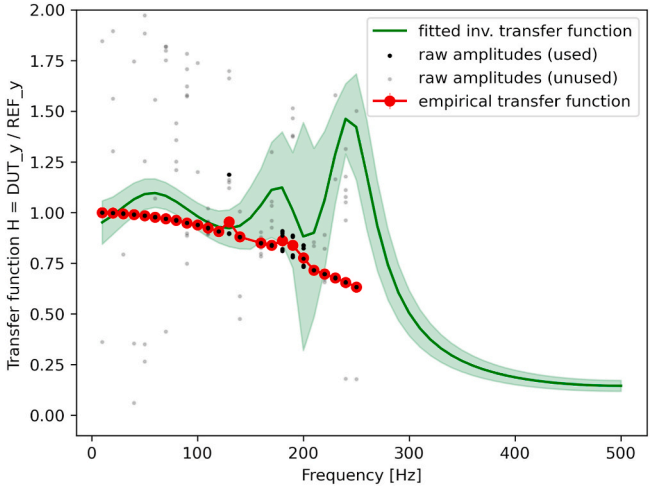

Fig. 9. Amplitude spectrum of empirical transfer behavior in DUT-y-direction (red) and fitted inverse transfer function (green). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

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

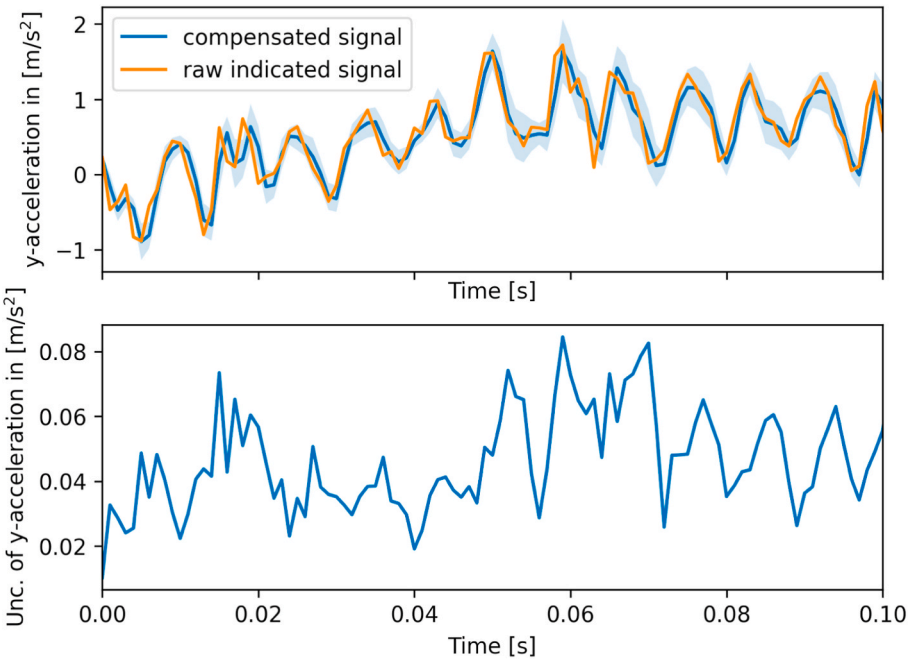

Fig. 10. Top: Comparison of indicated and compensated time series of the *y*-axis of the DUT (note: for better visualization an expanded uncertainty with $k = 5$ is shown). Bottom: Corresponding dynamic standard uncertainty of the compensated signal ($k = 1$).

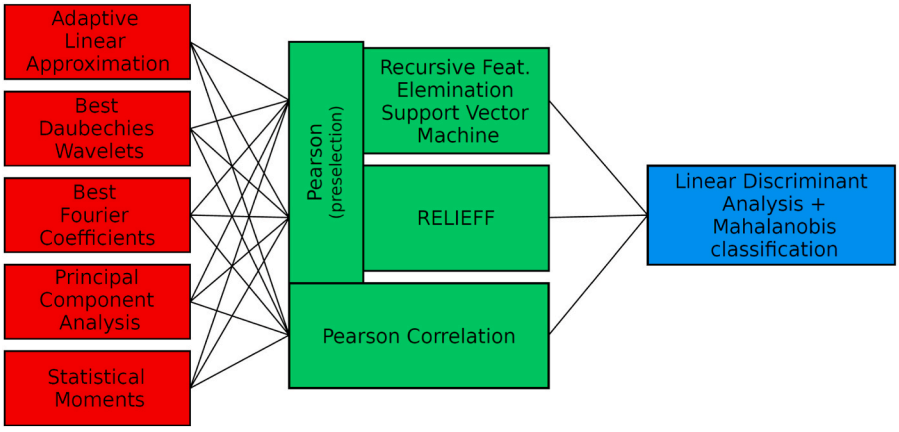

Fig. 11. Scheme of the ML toolbox with feature extraction (red), selection (green) and classification (blue) [30]. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

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.

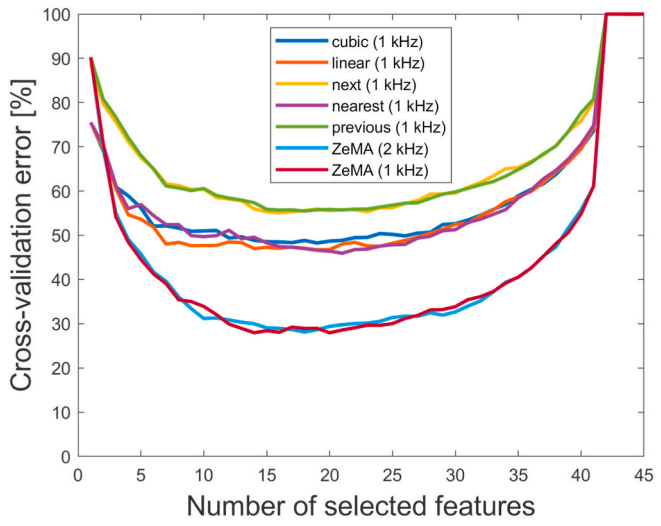

Fig. 12. Cross-validation performance of models trained on different input data sets. BFC is used as extractor and RFESVM as selector.

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.

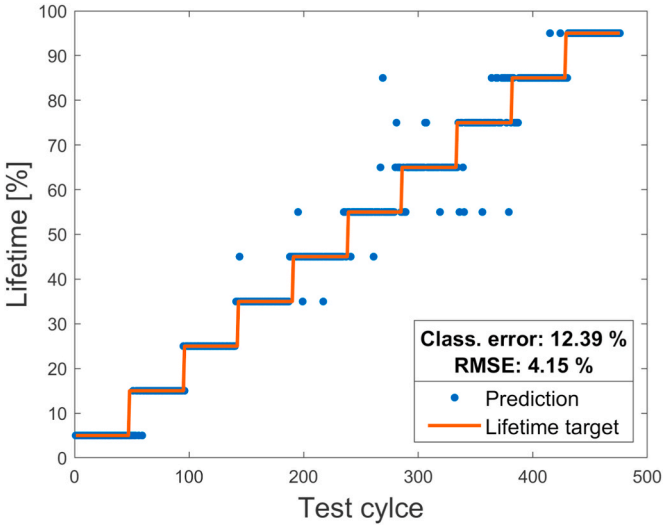

Fig. 14. Predicated lifetime of cubic interpolated SUU data compared with linear target (10% increments) for one fold of the 10-fold cross-validation.

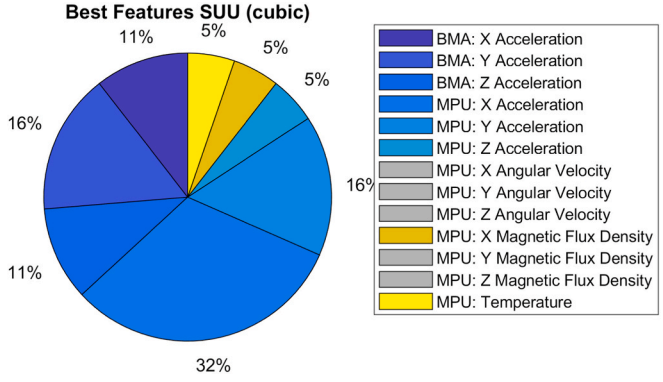

Fig. 15. Percentage of features selected from different sensors on the SUU that contribute to the lifetime estimate.

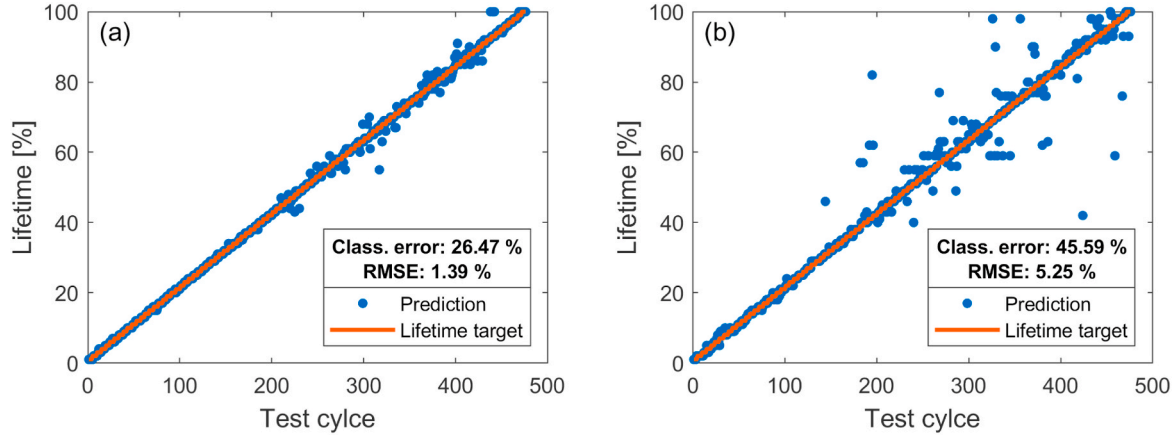

Fig. 13. Predicated lifetime compared with linear target (1% increments) for one fold of the 10-fold cross-validation. (a) 1 kHz ZeMA DAQ data and (b) 1 kHz cubic interpolated SUU data.

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))

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

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.

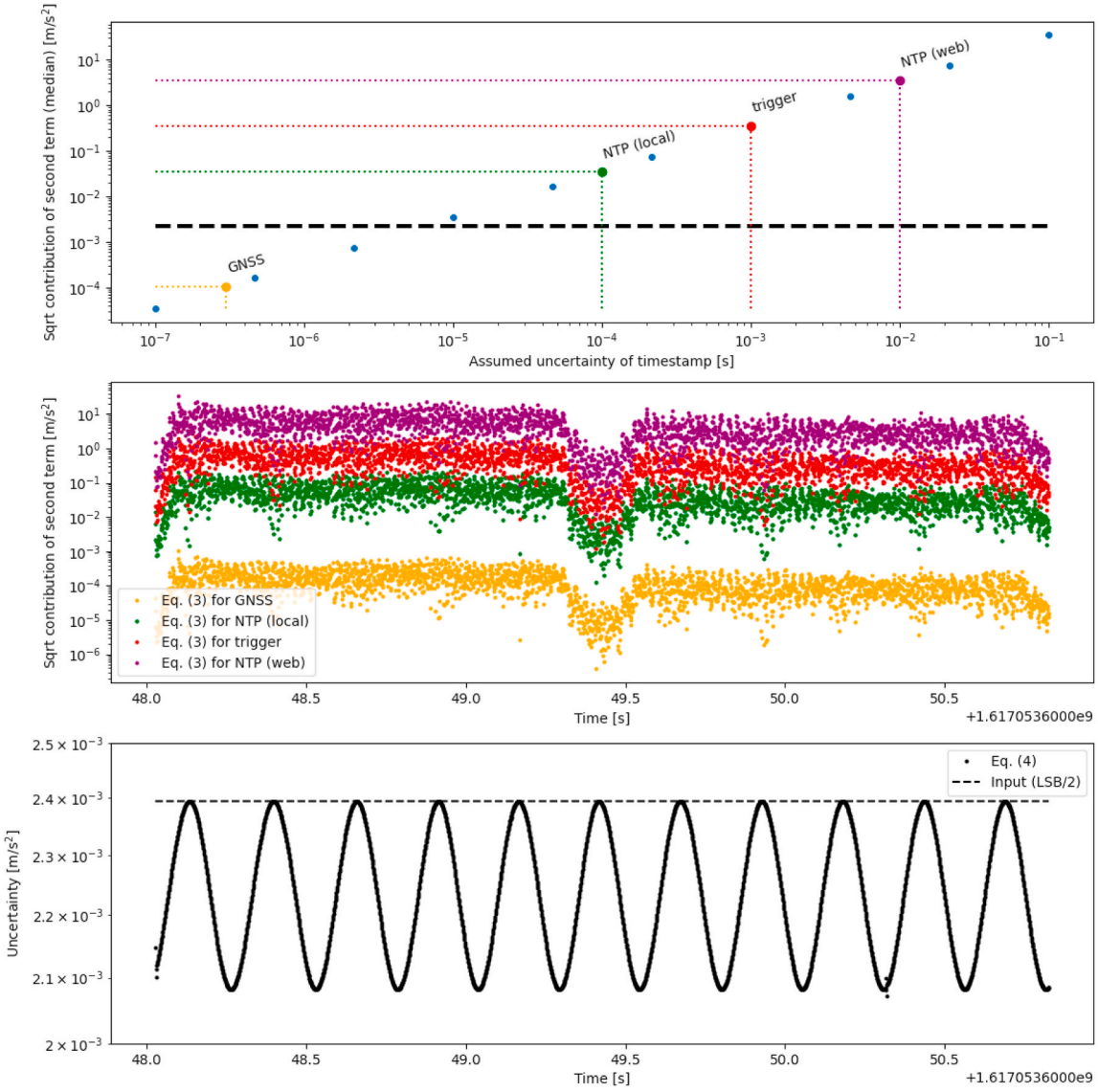

Fig. 16. Contribution of timestamp uncertainty onto interpolated data values for different assumed input time uncertainties.

Appendix B. Exemplary Raw Data

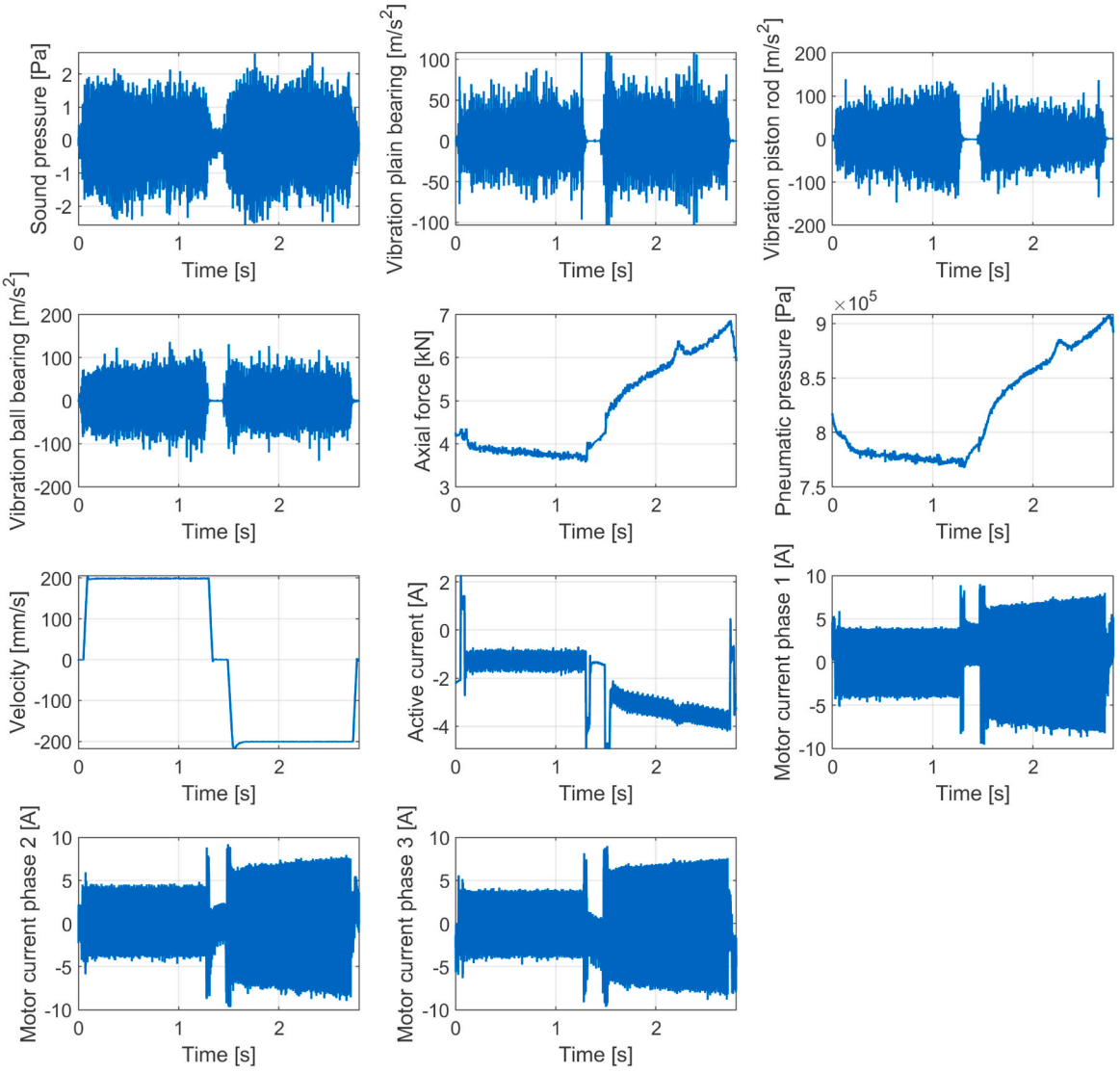

Fig. 17. Raw data recorded during the 51st cycle by the ZeMA DAQ expressed in SI units.

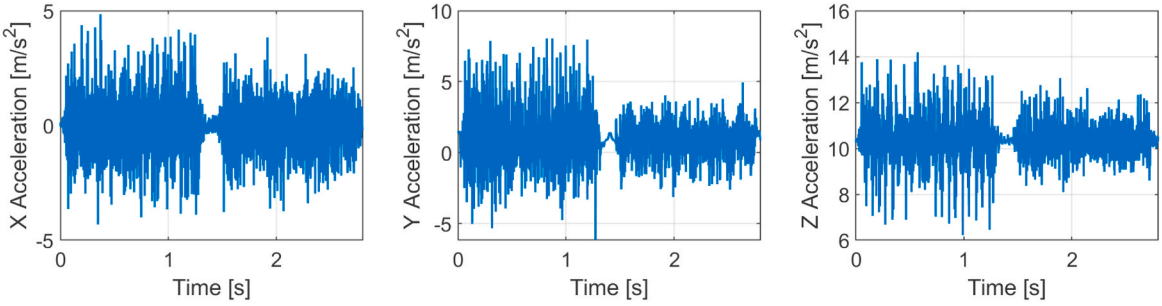

Fig. 18. Raw data recorded during the 51st cycle by the BMA 280 sensor of the SUU.

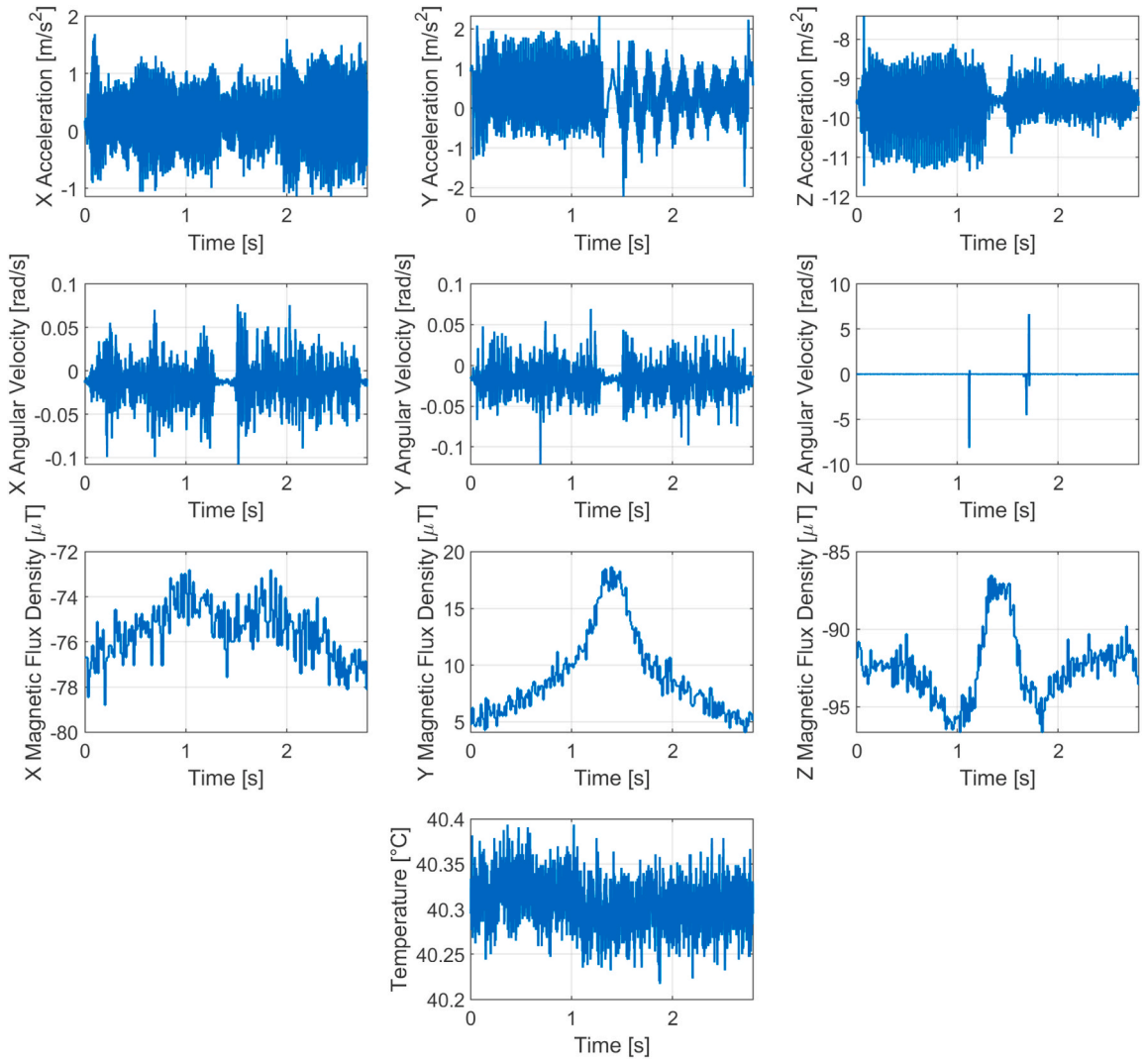

Fig. 19. Raw data recorded during the 51st cycle by the MPU 9250 sensor of the SUU.

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.