IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2024.0429000

Density-Based Dataset Balancing for Incremental Retraining in CNC Axis and Spindle Current Prediction

ROBIN STRÖBEL[1], AARON BÜTTNER[1], HAFEZ KADER[2], (Student Member, IEEE), ALEXANDER PUCHTA[1], BENJAMIN NOACK[2], (Senior Member, IEEE), and JÜRGEN FLEISCHER[1]

[1] wbk Institute of Production Science, Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany
[2] AMS - Autonomous Multisensor Systems, Otto von Guericke University Magdeburg, 39106 Magdeburg, Germany

Corresponding author: Robin Ströbel (e-mail: robin.stroebel@kit.edu).

ABSTRACT Accurate prediction of the axis and main spindle current is essential for reliable model-based process monitoring in Computer Numerical Control (CNC) machining. However, modern manufacturing is characterized by a high level of product variety, short life cycles, and frequently changing process conditions. This results in unbalanced and constantly changing data distributions, which present a challenge to model training. Traditional data collection strategies generate large and redundant datasets that require substantial computational and storage resources. This work introduces a density-based approach to dataset balancing for incremental retraining that enables efficient and robust model training under evolving operating conditions. The proposed approach operates at the data level via a density-controlled replay memory, rather than modifying model parameters. A reduced, interpretable feature representation is derived to represent the underlying data structure, providing a basis for systematic discretization. Based on this representation, the adaptive memory mechanism prioritizes samples according to their spatial density, selectively removing redundant information, and ensuring adequate coverage of regions critical for model performance. The proposed method has been evaluated across multiple machines, machining parameters, and geometries. The results show that the balancing strategy yields significantly better results than static training while reducing the required data volume by over 40% compared to incremental data expansion. Consequently, it facilitates the development of data-efficient, scalable and robust training strategies for intelligent machining systems, while also supporting improved adaptability in the monitoring of flexible production environments.

INDEX TERMS Machine tool, CNC, signal prediction, incremental retraining, data-centric AI.

I. INTRODUCTION

CNC milling machines are an essential part of modern manufacturing, enabling the production of complex components with great precision. Growing demand for customized products is driving machining environments towards one-off manufacturing [1]. This trend also affects individual aspects of production, such as process monitoring. To produce customized, high-quality, low-cost products, monitoring approaches also must be flexible and adaptable. Advances in sensor miniaturization and increased computational capabilities allow the collection of large volumes of high-resolution process data. In this context, machine learning (ML) has become indispensable, as it can identify complex relationships within extensive process data. Consequently, ML is becoming increasingly relevant in process modeling and monitoring [2].

Previous work introduced a hybrid ML model for predicting axis and spindle current signals, enabling residual-based anomaly detection [3]. Reliable residual generation requires model training on a dataset that represents the full range of relevant operating conditions. In practice, machining operations generate continuous data streams with evolving process states. Consequently, data distributions shift over time and datasets grow indefinitely, often containing substantial redundancy [4]. While some deep learning (DL) models can extrapolate, traditional ML models are fundamentally limited to the information contained in the training dataset [5]. This poses major challenges for storage, transmission, and retraining [6].

Therefore, the efficient management of these continuously expanding datasets is critical to maintain a representative and well-distributed database. To address this issue, this paper

proposes a density-based, data-centric balancing strategy for incremental retraining that constructs and maintains a compact and representative replay memory for the prediction of axis-specific current signals.

The remainder of this paper is organized as follows: Section II reviews the current state of the art in incremental learning (IL), data-centric AI, and data management approaches. Section III details the proposed methodology and the discretized data space. Section IV presents the experimental setup. The results are shown in Section V and validated in Section VI. Section VII concludes with a discussion of the findings. Section VIII summarizes the paper and suggests potential directions for future research.

II. STATE OF THE ART

IL provides an opportunity to learn from constantly expanding and evolving datasets. In this context, data-centric AI is increasingly important for maintaining high-quality training datasets. The following presents both directions and discusses the underlying paradigms and methods.

A. INCREMENTAL LEARNING

IL describes processes in which models acquire information over longer periods of time from sequentially arriving data while retaining knowledge that has already been learned. Such systems must adapt to newly available data without catastrophic forgetting, i.e., overwriting previously learned information, which is particularly relevant for Neural Networks (NN). Since data is often generated continuously in real-world applications, it is not possible to store all historical data in its entirety. Thus, IL has high relevance in real-world ML applications and addresses practical limitations such as limited storage capacity, continuous data streams, and changing data distributions. [7, 8]

IL can be divided into three groups: Task-IL, Domain-IL, and Class-IL [9]. In Task-IL, a set of clearly distinguishable tasks must be learned incrementally. It is always clear to the model which task is to be performed. This distinction between tasks means that completely separate models can be trained for each task. An example of task-IL is shown in [10] using various image datasets. The task is always known, so the model acts in a task-specific manner. A convolution NN architecture is given an additional encoder and channel for each new task (SILLY-N = Simple Lifelong Learning Networks). The special feature is that the channels are always connected to all encoders. This design allows forward transfer, as new channels use information from previous encoders, and backward transfer, as old channels can be updated by sending previous task data through the new encoders. For prediction, all encoders are used in parallel, their outputs are transferred via the task-specific channel, and then converted into a class decision by the decoder. Empirically, the proposed methods (SILLY-N and SILLY-F) consistently show positive forward and backward transfer values in numerous benchmarks, while competing methods typically fail in at least one of the two transfer directions.

The second scenario, Domain-IL, the task itself remains the same, but must be solved in different contexts [9]. In [11], an object classifier was trained on image data. The task of recognizing objects remained the same, but the domain in which the task had to be performed changed due to changing weather conditions, such as rain, snow, or sun. The Domain Incremental through Statistical Correction (DISK) approach presented is a method that does not retrain model parameters, but instead stores the batch normalization statistics (mean and variance) for each weather domain and uses them as needed. Since the weights of the model itself remain frozen, the approach enables zero forgetting across domains. Domain-IL is often of particular interest when the distribution of the underlying data changes [12].

In the third scenario, Class-IL, new additional classes are added incrementally and must be learned by the model. One broadly applicable strategy, which is suitable for various incremental scenarios, is described in [9]. This involves storing old data and using it for retraining in order to preserve old knowledge. In practice, there are examples that fall within the Class-IL scenario. In [13], a class-IL approach is presented that uses a representative memory for all classes. Through regular updates, the memory was repeatedly reweighted and used for training. The amount of data for the classes is also specified to ensure a balanced dataset. In [14], a class-IL system for image classification is presented, which stores class-representative images to represent a class.

Although there are many approaches to classification and image-based applications, IL for regression problems remains underexplored. While existing approaches focus on model-centric adaptations, the importance of the underlying dataset is increasingly recognized. This shift in perspective motivates data-centric approaches, which are introduced in the following section.

B. DATA-CENTRIC AI

The traditional training approach considers the training dataset to be a given, on which an ML model is then optimized. However, the systematic design and processing of data is crucial for the development of effective and efficient AI-based systems [15]. Consequently, data-centric AI emerges as a complementary paradigm focusing on training dataset optimization (see figure 1). Improvements in prediction quality are thus achieved by optimizing the database rather than through hyperparameter optimization of the model architectures.

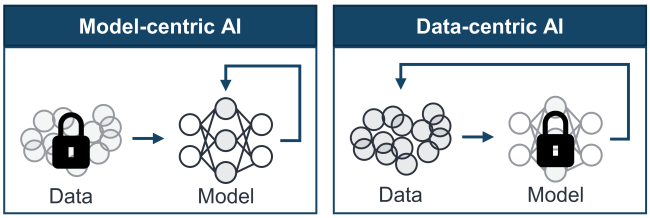


FIGURE 1. Model- and Data-centric AI

A comprehensive survey of deep learning model training is conducted in [16] with a focus on dataset optimization. Addressing underlying data issues is becoming an increasingly critical part of ML research. Under the newly emerging paradigm of data-centric AI, a sufficient amount of cleaned and prepared data is therefore seen as the basis for ML. This involves the steps of data collection, cleaning and validation, and finally robust model training. Data collection involves identifying and expanding suitable datasets using techniques such as data augmentation and labeling. Cleaning and validation focus on correcting erroneous data points and sanitizing the dataset. In [17], the authors argue that data-centric approaches must be supplemented with a repetitive update step. To maintain a solid, up-to-date database, it is crucial to continuously clean the database of erroneous data points, acquire specific data and augment data if necessary.

Unlike with classification tasks, it is more challenging to create a balanced dataset for regression models. However, targeted data augmentation represents a promising approach [18] to this challenge. [19] presents the geometric SMOTE algorithm for dataset balancing, which is used for targeted data augmentation in sparsely populated areas of the database. This augmentation is based on the database's distribution and uses a relevance function to give greater weight to areas where little data is available. The augmentation itself is performed using a hyper sphere stretched between rare real data points, offering greater variance than simple interpolation to counteract possible overfitting due to points that are too similar. The approach is shown to be more effective than random oversampling and other methods.

AL addresses the issue of having large amounts of unlabeled data available in many real-world applications, while annotation is expensive and time-consuming. The goal of AL is therefore to minimize the amount of labeled data required by selectively choosing particularly informative data points, while at the same time achieving a rapid reduction in generalization error [20]. Fundamentally, AL is based on an iterative process in which a model, initially trained on a small amount of labeled data, selects specific instances from a set of unlabeled data. The labels for these instances are requested from an oracle. These newly labeled instances are then added to the training dataset and the model is retrained [21].

There are three basic AL scenarios. The first is pool-based AL, where a closed set of unlabeled data is available. The second is stream-based AL, where data arrives sequentially as a stream. The third is AL based on membership query synthesis, where the learner generates new query instances based on the current model, often near the decision boundary. The Oracle annotates these instances. While this enables efficient model training, it can be problematic if instances are difficult for humans to interpret. [20, 22]

Upcoming approaches combine AL and augmentation for dataset optimization. [21] investigates a pool-based AL framework that combines uncertainty queries with data augmentation to further increase the efficiency of the learning process. In their approach, the most uncertain data points are selected from the unlabeled pool in each AL iteration, labeled, and added to the training dataset. In addition, artificial data points are generated using a modified variant of Geometric SMOTE (G-SMOTE) (see [19])to specifically improve both the amount of data and the data distribution.

A recent approach by [23] focuses on improving the training of deep NNs from a data-centric perspective. Rather than complicating model architectures and performing complex hyperparameter tuning, the focus is on improving the database. The data-centric approach involves creating a large, diverse and high-quality database. First, data quality is improved by removing duplicates and correcting incorrect and uncertain labels using an experimentally determined threshold. In a second step, data is augmented using contrast adjustments and translation. Classification experiments conducted with a ResNet-18 model showed that the data-centric approach outperformed the model-centric baseline by 3%. Current research provides practical examples of efficient data reduction [24] and intelligent data selection strategies for production-specific data. For instance, [25] proposes a sequential learning framework for defect classification in laser powder bed fusion. The approach uses an AL strategy in conjunction with uncertainty-based sampling. Synthetic points are distributed evenly in the feature space, and the uncertainty of these points is determined using a random forest model. Through AL, the real points with the lowest Euclidean distance to the most uncertain points are included in the training dataset reducing the required samples by 45%.

C. SUMMARY AND RESEARCH DEFICIT

IL addresses scenarios in which data accumulates gradually. Well-known approaches such as SILLY (Task-IL) [10] and DISC (Domain-IL) [11], as well as example-based methods such as iCaRL (Class-IL) [14], demonstrate that knowledge can be retained over long periods. However, these methods often rely on model-dependent architectures or specialized storage mechanisms to mitigate catastrophic forgetting at the parameter level and are primarily tailored to image classification tasks.

Meanwhile, the data-centric AI paradigm highlights the importance of high-quality, well-structured and representative data as a foundation for robust models. Research into dataset balancing, sampling strategies, and AL shows how datasets can be expanded or constructed selectively. Nevertheless, there is a lack of interpretable and data-efficient methods that can systematically reduce and balance industrial data streams. Data-centric approaches for IL are particularly beneficial for time series prediction models due to high computational cost of architecture selection and hyperparameter optimization. However, their application to regression tasks remains underexplored, and only few transferable solutions exist for continuously generated process data in manufacturing. Furthermore, the majority of existing methods focus on model-centric adaptations, whereas the role of systematic training data management is less well-explored. In particular, there is a lack of approaches that manage continuously

evolving datasets through the use of explicit data selection and replay mechanisms.

In the context of CNC process monitoring, this gap is particularly critical, since data is continuously generated and changes due to concept drift, resulting in the rapid accumulation of redundant data. Although IL approaches aim to preserve knowledge within the model, there is currently no systematic method of managing the underlying training data distribution in a structured and interpretable way. Consequently, existing approaches lack practical mechanisms for consistently reduce and balance the continuously growing volume of process data in a way that makes it both interpretable and suitable for the incremental retraining of a reference model. This raises the following research questions:

1) How can a reduced feature representation be constructed and discretized?
2) How can density-controlled replay memory be implemented within such a representation?
3) What is the effect of the density-controlled replay memory on retained data volume and prediction performance across incremental retraining iterations?

The main contribution of this work is a data-centric approach to managing continuously evolving datasets in the context of CNC process monitoring. It introduces a density-based balancing strategy for creating and maintaining a compact and representative replay memory for incremental retraining. Rather than modifying model parameters to address catastrophic forgetting, the proposed method operates at the data level via density-controlled replay, preserving the representativeness of the training data to maintain model performance. The validation in production scenarios across two machines demonstrates the effectiveness, and provides a methodological basis and a practical example for the data-efficient, scalable adaptation of models.

III. APPROACH/METHODOLOGY

The proposed approach is embedded within a residual-based monitoring framework (see figure 2). In such a system, a hybrid model [3] predicts axis-specific current signals of a CNC milling machine, which serve as reference values and are continuously compared with the measured current signal.

Unlike parameter-based IL approaches, the proposed method is data-centric and focuses on incremental retraining by controlling the distribution of training data over time. The aim is therefore to efficiently curate and maintain a compact, representative replay memory that supports robust model updates in changing operating conditions. Model improvements are therefore achieved by maintaining a robust and representative dataset, which serves as the foundation for complementary model-centric optimization.

To achieve this, incrementally acquired data is inserted into a structured data space that is divided into volumetric elements referred to as bins. These bins provide a discrete representation of the data distribution and form the basis of a density-controlled replay memory. Every bin is populated up to a predefined target density, yielding a balanced dataset

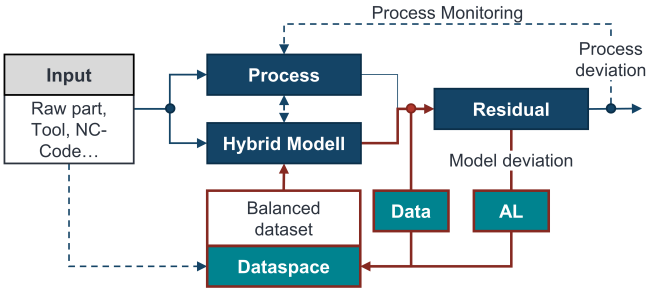


FIGURE 2. Incremental retraining loop (red and thick) as part of a residual-based process monitoring framework

over time as new operating conditions emerge. If model-induced deviations occur, the density in the relevant bins can be increased to enable targeted oversampling. However, oversampling is introduced as an interface for local data enrichment. A formal trigger criteria and a fully automated decision system are outside the scope of the present study.

The following section introduces the underlying hybrid model and its ML components. Based on this, a reduced feature representation is derived, and the approaches to its discretization and the density-based data balancing are presented.

A. HYBRID MODEL AND INITIAL FEATURE SPACE

In order to perform feature-space reduction, the input variables of the underlying ML model are introduced. The methodology is based on the axis-level current signal prediction framework presented in [3]. It combines kinematic machine data, including axis position, velocity, and acceleration, with a process simulation (see figure 3). The simulation derives signals that are relevant but difficult or expensive to measure, such as cutting forces or the material removal rate (MRR). These values are calculated according to [3]. These signals then serve as the input for ML models specific to each axis, which output the current signal of the drives.

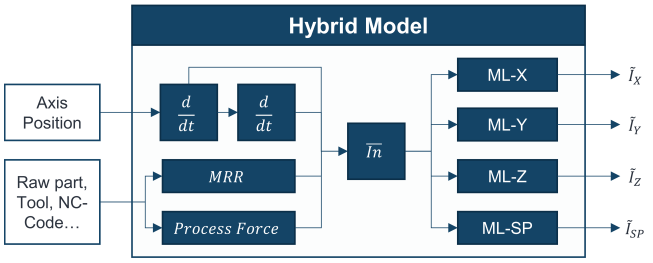


FIGURE 3. Hybrid model structure for reference signal prediction [3]

In the case of a three-axis milling machine, the hybrid model structure can be described by axis- and spindle-specific ML models

$$Out_n = M_n(\overline{In}) = \tilde{I}_n, \text{ where } n \in \{x, y, z, sp\} \qquad (1)$$

which map the input vector

$$\overline{\text{In}} = \left[v_x, v_y, v_z, a_x, a_y, a_z, \dot{\varphi}, \ddot{\varphi}, F_x, F_y, F_z, M_{sp}, MRR\right]^T \quad (2)$$

to the respective target $\tilde{I_x}$, $\tilde{I_y}$, $\tilde{I_z}$, and $\tilde{I_{sp}}$.

B. FEATURE SPACE REPRESENTATION FOR BALANCING

The original feature space is complex and high-dimensional due to its 13 dimensions. This makes it difficult to describe and manage data accumulations and their relationships efficiently. Moreover, a 13-dimensional space is difficult for human operators to comprehend. Therefore, the reduced feature representation aims to describe the data based on a smaller set of interpretable dimensions, providing a foundation for data organization and dataset balancing. With three dimensions, a sufficient amount of information can be included while maintaining intuitive clarity. It is important to note that the reduced feature representation is used exclusively for data organization and selection. The ML models continue to operate on the full 13-dimensional input space defined in formula 2. Consequently, the reduced feature representation does not affect the information available for prediction, as the original feature space is fully preserved during model training.

A feature importance analysis in [3] showed that axis velocity is highly important, enabling the movement distribution to be decoupled from the current tool center point position. In planar milling operations, the axis speeds v_x and v_y represent the feed velocity, an important and easily understandable variable in the milling process. The MRR is particularly well-suited to serve as the third dimension, as it correlates with many other factors, including technological parameters and cutting forces. Additionally, [3] showed that MRR is one of the most important features for model prediction. Another advantage is the cross-axis information that is bundled in MRR. To separate the speed component from the MRR, which is already represented in the v_x/v_y-plane, the tool engagement area

$$\text{AR} = \frac{\text{MRR}}{v_c}, \qquad \text{with} \qquad v_c = \sqrt{v_x^2 + v_y^2} \quad (3)$$

is defined as the third dimension. Figure 4 shows data distributions in the resulting space at different cutting depths and feed velocity. As can be seen, datasets consisting of movement in all directions at a given feed velocity and cutting depth lead to ring-shaped distributions.

The radial arrangement of the data in this space enables a description in cylindrical coordinates with

$$\begin{aligned} x &= v_c \cos(\varphi), \\ y &= v_c \sin(\varphi), \\ z &= AR. \end{aligned}$$

In this reduced feature representation $\mathcal{M} = (v_c, \varphi, AR)$, a 13-dimensional data point can be represented as a three-dimensional tuple. This representation is a structured projection of the original feature space for density estimation rather than a replacement for the model input space. Consequently, the reduced feature representation enables efficient and interpretable organization of the data for management purposes, while the predictive model itself retains access to the full input space.

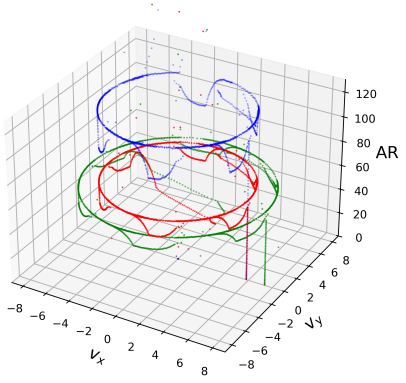


FIGURE 4. **Gear part in the reduced representation** at various feed velocities and cutting depths

C. DISCRETIZATION AND DEFINITION OF DENSITY

The parameters of memory $\mathcal{S}$ result from the discretization of $\mathcal{M}$ into volume elements, referred to as bins. The reduced feature representation $\mathcal{M}$ is divided into b discrete areas

$$B_{ijk} = \left\{ (v_c, \varphi, AR) \in \mathbb{R}^3 \;\middle|\; \begin{array}{c} v_c \in [\xi_i, \xi_{i+1}), \\ \varphi \in [\eta_j, \eta_{j+1}), \\ AR \in [\zeta_k, \zeta_{k+1}) \end{array} \right\} \quad (4)$$

via the interval limits ξ_i, η_j and ζ_k. v_c represents the feed velocity in $[mm/s]$, φ represents the mathematically positive angle to the positive x-axis, and AR represents the tool engagement area in $[\frac{\text{mm}^3/\text{s}}{\text{mm/s}}] = [\text{mm}^2]$. Each bin has a unique index $I_{v_c,\varphi,AR}$ that depends on its position within $\mathcal{M}$. Each data point can be assigned to a bin and thus has an index. The resulting memory $\mathcal{S}$ can be seen in figure 5.

The reduced feature representation $\mathcal{M}$ can now be described via discrete elements (see figure 6). Therefore, all data points within a bin are considered similar. The volume of a bin and the number of data points it contains can be used to calculate a density, which controls the amount of data permitted within a bin. The volume of a bin is calculated by integrating across its boundaries in cylindrical coordinates. For an isolated volume element

$$dV = \int_{\zeta_k}^{\zeta_{k+1}} \int_{\xi_i}^{\xi_{i+1}} \int_{\eta_j}^{\eta_{j+1}} v_c \, \mathrm{d}\varphi \, \mathrm{d}v_c \, \mathrm{d}AR \quad (5)$$

and

$$V_{\text{Bin}} = v_{\text{c,center}} \cdot \Delta\varphi \cdot \Delta v_c \cdot \Delta AR \quad (6)$$

follows. The value $v_{\text{c,center}}$ describes the center of a bin in the direction v_{c} The density

$$\rho_{\text{Bin}} = \frac{N_{\text{Bin}}}{V_{\text{Bin}}} \quad (7)$$

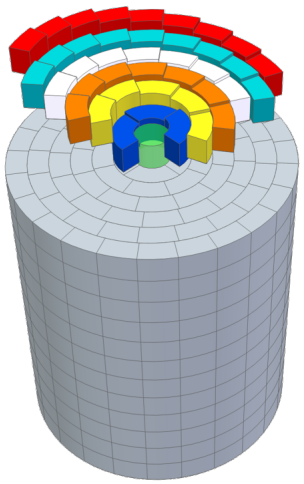


FIGURE 5. Structure of memory $\mathcal{S}$, which discretizes the reduced feature representation

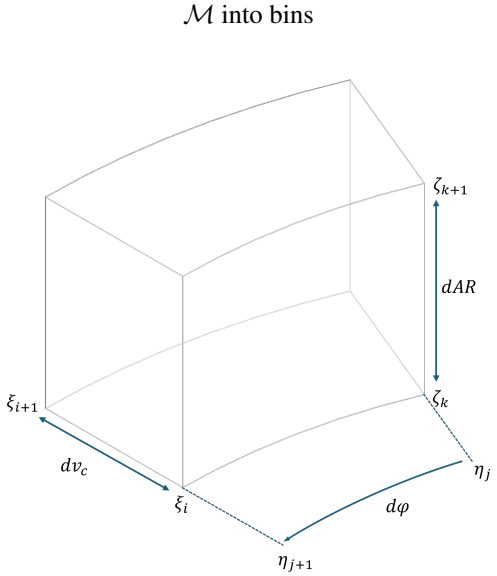


FIGURE 6. One Bin of memory $\mathcal{S}$

is calculated from the volume of a bin and the number of data points N it contains. The density thus controls the maximum number of points allowed per bin. $\mathcal{M}$ represents less information than the initial feature space due to the reduction. Therefore, local oversampling can be beneficial.

Another important aspect of the memory $\mathcal{S}$ is its data clearing function which guarantees that only data within the limits of the reduced feature representation $\mathcal{M}$ and thus originating from the manufacturing process is stored.

$$\text{MRR} > 0; \quad v_{\text{min}} \leq v_{\text{c}} \leq v_{\text{max}}; \quad AR < AR_{\text{max}} \tag{8}$$

Thus, events like main spindle ramp up and movements outside the material are filtered. Algorithm 1 describes the steps of the initialization of memory $\mathcal{S}$.

Algorithm 1 Construction of Memory $\mathcal{S}$ in $\mathcal{M} = (v_c, \varphi, AR)$

Input: $v_{\text{max}}, AR_{\text{max}}, V_{\text{bin}}, \Delta v_c, \Delta AR$
Output: df_{bins}
Indexes $\leftarrow []$
$\text{Index}_{v_c} \leftarrow \text{arange}(0, v_{\text{max}} + \Delta v_c, \Delta v_c)$
$\text{Index}_{AR} \leftarrow \text{arange}(0, AR_{\text{max}} + \Delta AR, \Delta AR)$
for (AR_{idx}, AR) **in** Index_{AR} **do**
  for $(v_{c\text{idx}}, v_c)$ **in** Index_{v_c} **do**
    $v_c \leftarrow \text{Index}_{v_c}[v_{c\text{idx}}]$
    $v_{c,\text{mid}} \leftarrow v_c + \frac{\Delta v_c}{2}$
    $\Delta\varphi^* \leftarrow \frac{V_{\text{bin}}}{v_{c,\text{mid}} \cdot \Delta v_c \cdot \Delta AR}$
    $n_\varphi \leftarrow \lfloor 2\pi / \Delta\varphi^* \rfloor$
    if $n_\varphi < 1$ **then**
      $\Delta\varphi \leftarrow 2\pi$
      $n_\varphi \leftarrow 1$
    else
      $\Delta\varphi \leftarrow 2\pi / n_\varphi$
      $\varphi_{\text{bins}} \leftarrow \text{linspace}(0, 2\pi, n_\varphi + 1)$
    end if
    Indexes.append($\{z_{\text{idx}}, r_{\text{idx}}, v_c, v_{c,\text{mid}}, n_\varphi, \varphi_{\text{bins}}\}$)
  end for
end for
$\text{df}_{\text{bins}} \leftarrow \text{DataFrame}(\text{Indexes})$
return df_{bins}

IV. EXPERIMENTAL SETUP

A. MACHINE AND DATASETS

The machining processes were recorded at 500Hz on a DMC 60H from the manufacturer DMG Mori [26]. The dataset includes several parts of the components Gear (G), Notch (N) and Plate (P) milled with a 10 mm end mill and a base a_p of $6mm$ and a_e of $10mm$, which can be seen in figure 7. The parts were manufactured in several series with varying feed rates F and cutting depths a_p. Due to planar milling, the speed in the z-direction is $v_z = 0$ during milling.

For steel S235 JR, F is $576mm/min$ and S is $3200rpm$, while for aluminum AL 2007 T4, F is $350mm/min$ and S is $3800rpm$. These parameters represent base manufacturing parameters or medium setting. In addition, a_p ranges from 3 to $12mm$. For aluminum F varies from 267 to $442mm/min$ and S from 2900 to $4800rpm$. For steel, F varies between 432 and $720mm/min$ and S between 2400 and $4000rpm$. The following refers to these machining parameter settings as 'high' and 'low'. The dataset contains 5x G_{Alu}, 5x P_{Alu} and 3x N_{Alu} parts, as well as 5x G_{Steel} and 5x P_{Steel} parts manufactured in the base parameter setting. To obtain a comprehensive dataset for experiment 1, additional high and low settings for S/F and a_p and three blowhole datasets are added for the Gear and Plate component in both Materials.

Based on [3], an ExtraTrees architecture is used for the X, Y, and Z axes (estimators = 30, max depth = 100, leafs = 5, nodes = 1000), and a Gradient Boosting model is used for the SP axis (estimators = 100, nodes = 6000). However, the approach presented is model-agnostic and can be used with various ML models depending on the problem at hand.

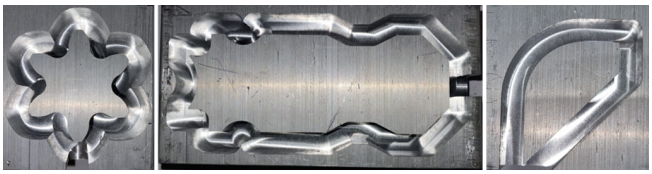
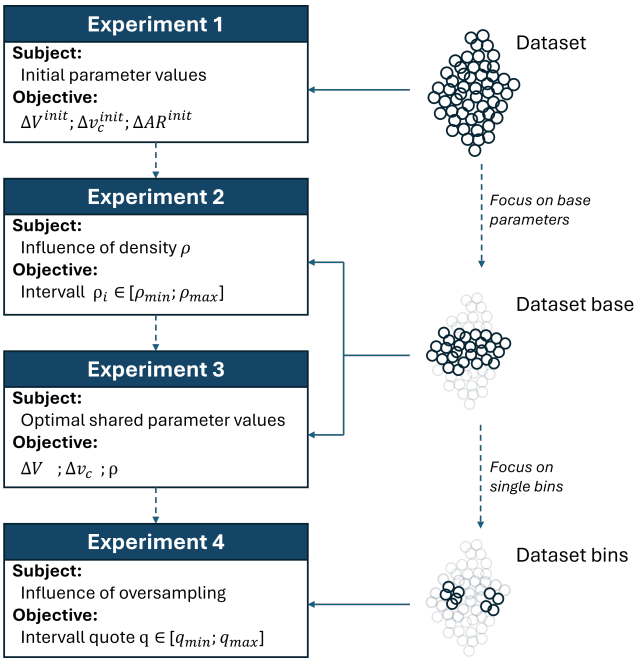


FIGURE 7. Gear, Plate and Notch from [26]

FIGURE 8. Experimental setup



Similar to [3], the dataset has been downsampled to 50 Hz.

Figure 8 shows the structure of the experiments, which are described in the following. Experiments 1 to 4 are conducted in succession to form a structured parameter study, defining the discretisation and density of the reduced feature representation.

B. EXPERIMENT 1: INITIAL PARAMETERS

The first experiment determines the initial parameters as the starting point for the subsequent investigations. The discretization of the feed velocity and the tool engagement is performed using a cube-root binning heuristic, where the number of bins is proportional to $\sqrt[3]{n}$ and the bin width is defined as the data range divided by the number of bins. This yields material-specific initial parameters $\Delta v_{c,material}$ and $\Delta AR_{material}$ for aluminum and steel.

C. EXPERIMENT 2: DENSITY INVESTIGATION

Experiment 2 evaluates the influence of the maximum bin density $\rho_{\max}$ on the prediction quality based on the parameters identified in Experiment 1. To ensure a consistent filling of the reduced feature representation $\mathcal{M}$, the base datasets of both materials are used. As these fall within the same region of $\mathcal{M}$, they are best suited to fill as many bins as possible. Starting from this value, the density is progressively reduced following a geometric decay (factor 0.8) down to a minimum of 3. The resulting ρ_i are shown in table 1.

TABLE 1. Densities ρ_i for aluminum and steel in experiment 2

Material	density ρ_i
Aluminium	2086, 1668, 1335, 1068, 854, 683, 546, 437, 350, 280, 224, 179, 143, 114, 91, 73, 58, 46, 37, 30, 24, 19, 15, 12, 9, 7, 6, 5, 4, 3
Steel	350, 280, 224, 179, 143, 114, 91, 73, 58, 46, 37, 30, 24, 19, 15, 12, 9, 7, 6, 5, 4, 3

For each level, the training data is limited to a maximum of $\rho_{\max}$ samples per bin. Models are trained on data from memory S and tested on the remaining dataset of the same material. Each configuration is repeated 30 times to ensure robustness. The objective of this experiment is not to determine a single optimal density, but to characterize its influence and to identify a suitable range for ρ_i to guide experiment 3.

D. EXPERIMENT 3: OPTIMAL PARAMETERS

Experiment 3 aims to identify the optimal parameter configuration for V_{Bin}, Δv_c and $\rho_{\max}$. Since all experiments are conducted under full tool engagement, ΔAR is kept fixed to reduce computational complexity. Although the parameter $d\varphi$ is not explicitly examined, it is indirectly included based on the bin volume formula. The parameter space is explored using Optuna [27], which enables an efficient identification of suitable parameter combinations. A total of 250 trials are performed, with each V_{Bin}- Δv_c - $\rho_{\max}$ configuration evaluated over 30 repetitions to ensure robustness. The investigated parameter ranges are summarized in Table 2.

TABLE 2. Parameters of experiment 3

Parameter	Aluminium	Steel
Δv_c	$0.1 \ldots 0.5$, step $= 0.1$	$0.1 \ldots 2$, step $= 0.1$
V_{Bin}	$0.35 \ldots 2$, step $= 0.02$	$0.35 \ldots 2$, step $= 0.01$
$\rho_{\max}$	37, 46, 91, 114, 143, 179, 193, 224, 280, 350, 437, 546, 683, 854, 1068, 1335	37, 46, 91, 114, 143, 179, 193, 224
ΔAR	3.2	3.8
tests	$5 * 82 * 16 = 6560$	$5 * 165 * 8 = 6600$

The objective is to identify configurations that yield robust performance across both materials, enabling a uniform and transferable memory structure.

E. EXPERIMENT 4: OVERSAMPLING INTERFACE

Since $\mathcal{S}$ is organized in the reduced feature representation $\mathcal{M}$, it may not capture all distinctions needed to identify informative samples. If prediction quality remains poor despite bins reaching $\rho_{\max}$, targeted oversampling may be beneficial. Experiment 4 therefore evaluates targeted oversampling as a proof-of-concept enrichment interface. A fully automated decision system with formal trigger criteria is outside the scope and remains future work.

The experiment considers bins filled up to $\rho_{\max}$ that share the same index across different components (see figure 9). For

two components, A and B, the oversampling of bins with identical indices $I^A_{\mathrm{AR},v_c,\varphi} = I^B_{\mathrm{AR},v_c,\varphi}$ is investigated. Additional data from component A is added to the corresponding bin of B to evaluate whether missing information can be compensated by targeted oversampling. For each oversampling ratio, the model is trained using enriched bin data and evaluated on the corresponding component. To assess potential side effects, the model is also evaluated using unused data from the original component with the same bin index. Each experiment is repeated 30 times to obtain robust results.

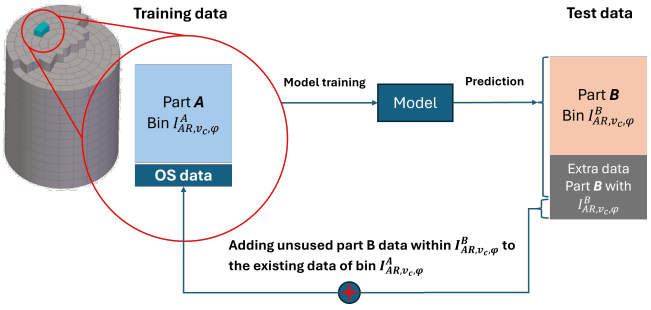


FIGURE 9. **Oversampling procedure for experiment 4**

F. EVALUATION METRICS

The mean squared error (MSE) is used since it represents a common performance metric for regression models. It is calculated as the sum of the squared residuals

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 . \tag{9}$$

A low MSE means a higher prediction quality on a given time series. To evaluate overall performance across all a axes, the sum

$$\mathrm{MSE}_{\mathrm{total}} = \sum_{\mathrm{axis}}^{A}\mathrm{MSE}_a = \mathrm{MSE}_X + \mathrm{MSE}_Y + \mathrm{MSE}_Z + \mathrm{MSE}_{SP} \tag{10}$$

and $\mathrm{MSE}_{\mathrm{mean}} = \frac{1}{A}\cdot\mathrm{MSE}_{\mathrm{total}}$ are given. The total number of axes a is indicated by A, which in this case is equal to 4. Furthermore, R^2

$$R^2 = 1 - \frac{1}{n}\cdot\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{11}$$

is used. To determine the performance across all four axes, it is divided by A

$$\mathrm{R}^2_{\mathrm{total}} = \frac{1}{A}\cdot\sum_{\mathrm{axis}}^{A}\mathrm{R}^2_a . \tag{12}$$

V. RESULTS

A. EXPERIMENT 1: INITIAL PARAMETERS

To define the initial parameters and constrain the search space, the reduced feature representation $\mathcal{M}$ with its filtering effect is applied (see Eq. 8). Based on the NC-Code, the maximum values are $v_{\max} = 12mm/s$ and $AR_{\max} = 120mm^2$. Considering only material-removing operations (MRR>0) and applying a safety margin, these limits are extended to $v_{\max} = 13mm/s$ and $AR_{\max} = 130mm^2$. The cube-root discretization yields 41 and 34 bins for aluminum and steel, respectively. The resulting histograms are shown in figure 10.

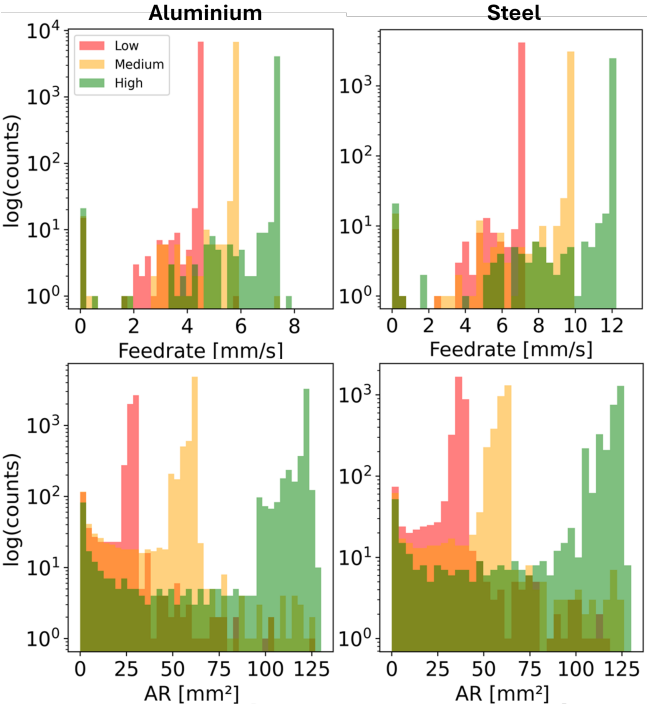


FIGURE 10. **Histograms of varying v_c and AR for the datasets**

The histograms reveal distinct operating regimes, confirming that the selected discretization is sufficient to separate relevant process states for both v_c and AR. This yields initial parameters $\Delta v_{c,al} = 0.2mm/s$, $\Delta AR_{al} = 3.2mm^2$, $\Delta v_{c,st} = 0.4mm/s$ and $\Delta AR_{st} = 3.8mm^2$. To complete the bin definition, a third dimension is introduced via the bin volume. It is initialized as $V_{\mathrm{Bin}} = 1.65\frac{mm^4}{s^2}$, corresponding to an angular division of approximately 10 degrees at 6 mm/s for steel. This fixed bin size is used in Experiment 2 to isolate the effects of the data density.

B. EXPERIMENT 2: DENSITY INVESTIGATION

Based on the initial parameters, the influence of different densities ρ_i on the prediction performance is examined for both materials. Figure 11 shows an asymptotic relationship between density and prediction performance for steel. Both MSE_{total} and R^2_{total} improve with increasing density and show diminishing returns at a density beyond approximately 37.

An analysis of individual axis indicates that this behavior is not uniform across all axes. While some follow the overall trend closely, others exhibit different sensitivities. This is illustrated by the Z-Axis (figure 12), which shows a pro-

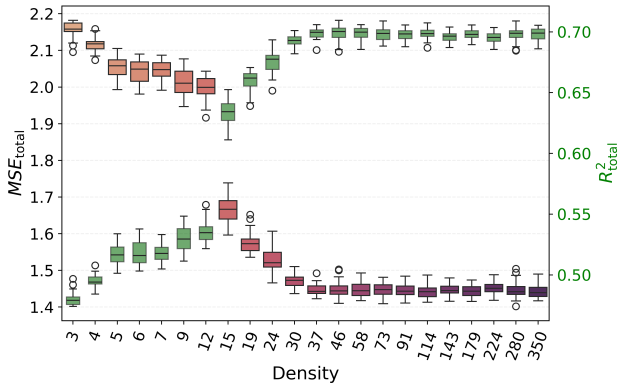


FIGURE 11. MSE_{total} and R^2_{total} across different densities (steel)

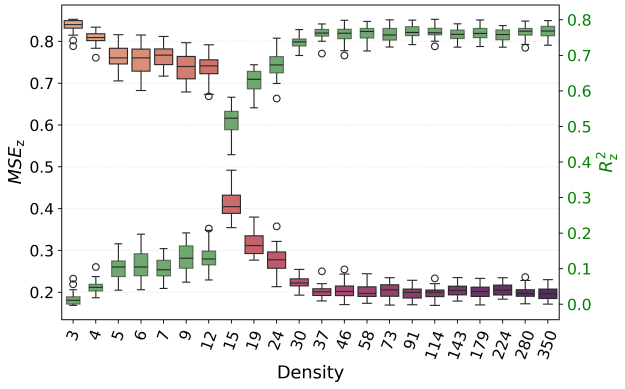


FIGURE 12. MSE and R^2 for the Z-axis (steel)

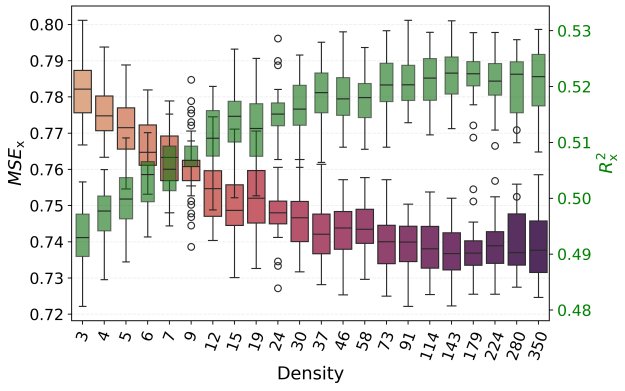


FIGURE 13. MSE and R^2 for the X-axis (steel)

nounced saturation effect, in contrast to the X-Axis (figure 13), where the influence of density is comparatively weak.

A similar asymptotic behavior is observed for aluminum at a higher density level of approximately 143 (see figure 14). Compared to steel, the influence of data density is more pronounced on the interpolated axes and the main spindle (see Appendix, figure 34 and 35).

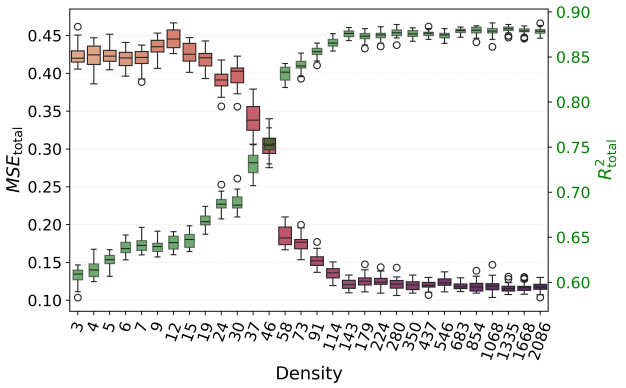


FIGURE 14. MSE_{total} and R^2_{total} different densities (aluminum)

C. EXPERIMENT 3: OPTIMAL PARAMETERS

Based on the density study, Experiment 3 determines suitable combination of parameters for Δv_c, V_{Bin} and ρ. Since *MSE* and R^2 show comparable trends across the individual axes, the analysis is carried out using MSE_{total}.

For both steel and aluminum (figure 15 and 16), MSE_{total} remains largely stable across the search space, but increases sharply for low-density settings, especially when combined with small $\Delta v_c \in \{0.1, 0.2\}$. In contrast, V_{Bin} spans a wide range among both good and poor configurations, indicating that ρ and Δv_c are the dominant drivers of performance degradation. This behavior is expected because ρ limits the number of samples per bin. Since $\Delta\varphi$ is set implicitly to satisfy equation 6, smaller Δv_c can result in large $\Delta\varphi$ boundaries, which can blur distinctions in the reduced feature space.

To obtain a transferable parameter basis, a combined optimization is conducted using an adjusted search space derived from the previous results (see Appendix, Table 7). Since V_{Bin} showed comparatively weak influence, its range is retained.

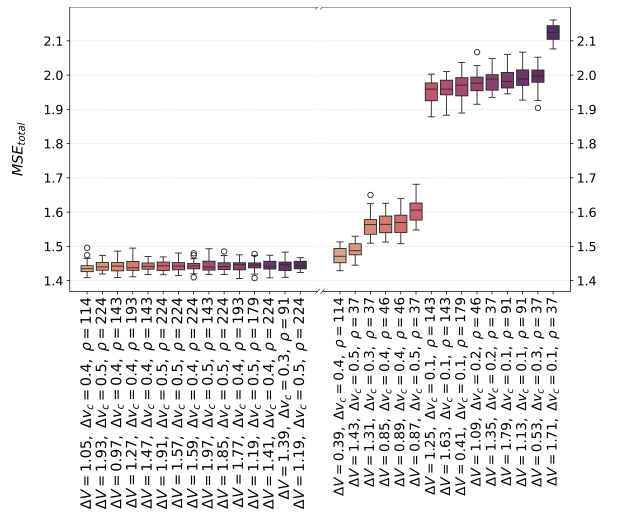


FIGURE 15. Minimum and maximum MSE_{total} for steel

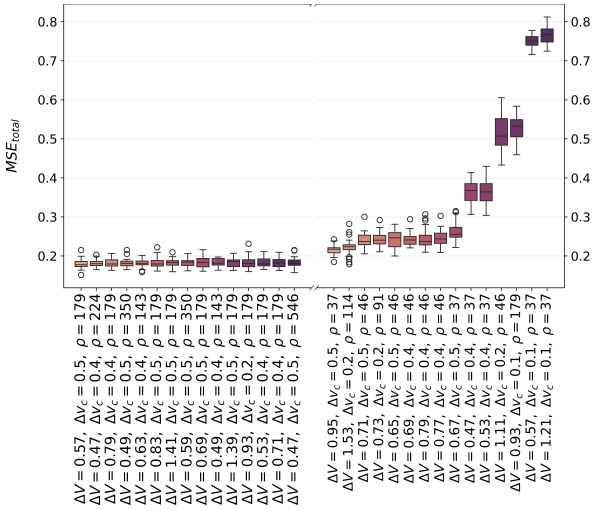


FIGURE 16. Minimum and maximum MSE_{total} for aluminum

In contrast, the low-ρ/low-Δv_c region associated with low performance is excluded by increasing the respective lower bounds. ΔAR is set to 3.5, and optimization is performed with 250 Optuna trails. To derive a unified, transferable parameter set across materials, the best performing configurations obtained for steel and aluminum are averaged (details in the Appendix, Table 8). The resulting parameters are $\Delta v_c = 0.5$, $V_{\text{Bin}} = 1.3$, $\rho_{\max} = 240$ and $\Delta\text{AR} = 3.5$. Figure 17 illustrates the resulting discretized relay memory $\mathcal{S}$. This setting results in a minor increase in MSE_{total} compared to the material-specific optima, while substantially simplifying deployment across materials.

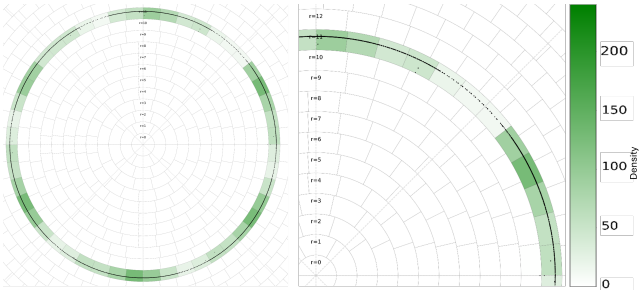


FIGURE 17. Aluminum Gear parts with base parameters in the Memory $\mathcal{S}$

D. EXPERIMENT 4: OVERSAMPLING INTERFACE

Based on the optimal memory parameters, Experiment 4 investigates targeted oversampling as a proof-of-concept interface. Figure 18 shows the memory for the notch and plate series, which share bins reaching $\rho_{\max}$. Table 3 summarizes the available data and the corresponding bins. Each bin is limited to $\rho_{\max} = 240$, corresponding to 317 data points, while additional samples can be used for oversampling. Red-circled bins enable oversampling up to a ratio of 2.0, while yellow-circled bins allow lower ratios.

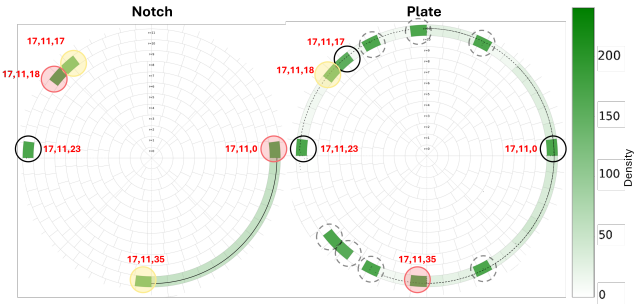


FIGURE 18. Maximum density of bins filled by aluminum part data

TABLE 3. Data of the Plate and Notch series

Index	Plate				Notch			
	in Bin	Density	Unused	Total	in Bin	Density	Unused	Total
1 (17,11,0)	317	240	52	369	317	240	375	692
2 (17,11,17)	287	217	–	287	317	240	144	461
3 (17,11,18)	317	240	195	512	317	240	512	829
4 (17,11,23)	317	240	15	332	317	240	3	330
5 (17,11,35)	317	240	2058	2375	317	240	190	507

To analyze the effect of oversampling, plate bins are used for training, while the corresponding notch bins are used for oversampling. Figure 19 shows representative results for bin 1 and 3 up to an oversampling ratio of 2.0. Oversampling has a positive effect on target predictions, resulting in a decrease in MSE_{total}. At the same time, variability increases slightly at higher oversampling ratios. Additional results for bin 1, 2, 3 and 5 up to a ratio of 1.45 are provided in the Appendix (figure 36). When considering the influence on the origin bins, an opposite effect is observed. Figure 20 shows that higher oversampling ratios can increase MSE_{total}. However, this deterioration remains smaller than the improvement on the notch data. This effect is even less pronounced for plate origin bins 1, 3 and 5 (see Appendix, figure 37). The reverse direction (notch-to-plate) shows similar behavior and is reported in the Appendix (figure 38-41). Oversampling reduces MSE_{total} for the target plate bins, while the origin notch bins show an upward trend starting at a ratio of approximately 1.5.

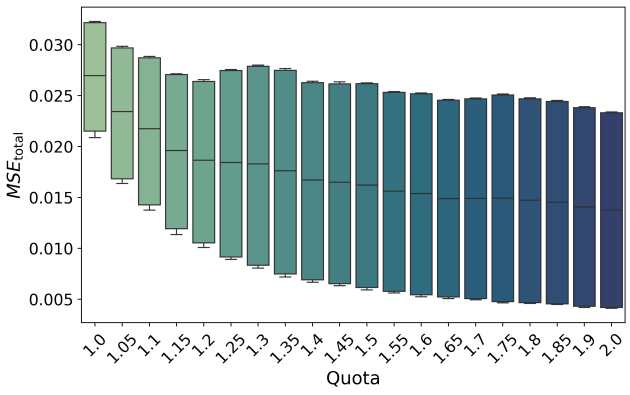


FIGURE 19. MSE_{total} for notch bin 1 and 3

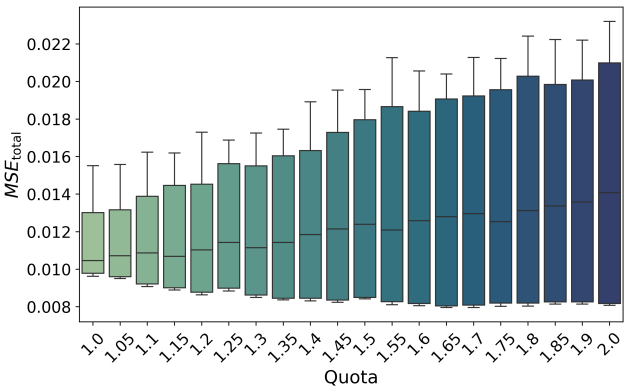


FIGURE 20. MSE_{total} **of the plate origin bins 1 and 3**

Overall, the results suggest that targeted oversampling can improve prediction performance when relevant information is not fully covered by $\mathcal{M}$. Although high oversampling ratios may slightly increase MSE_{total} for the origin component, this effect is outweighed by the improvement in the target component. The most effective oversampling ratio is observed between approximately 1.3 and 1.6. In safety-critical monitoring applications, however, the potential performance degradation for the origin component should be considered.

VI. VALIDATION

The aim of the validation is to demonstrate the potential of the density-based balancing approach by applying it to different machines, parts, and technological parameters.

A. DATASETS AND SETUP

As the focus is now on investigating incremental storage during the production of a series of components, adequate datasets are required. Here, we refer to the component series of an impeller, which was manufactured from POM-C on a DMC 60H milling machine (see [28], 12 parts) and a CMX 600V three-axis milling machine (see [29], 11 parts). The parts were created using CAD and an NC code was derived via CAM, giving them realistic and complex contours. A detailed description of the series is available in [29]. Figure 21 shows exemplary parts of the impeller series.

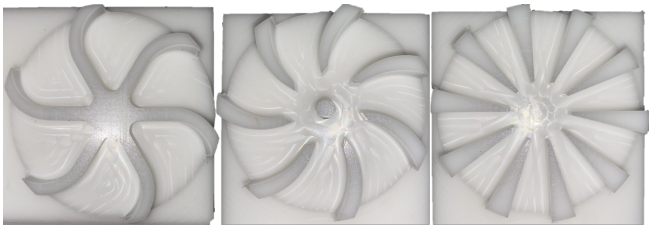


FIGURE 21. **Exemplary parts of the impeller dataset**

There is increased complexity in both setups due to the changed tool path and material (POM-C), resulting in new technological parameters (tool, feed rate, spindle speed, etc.). Moreover, the radial cutting depth varies due to the complex

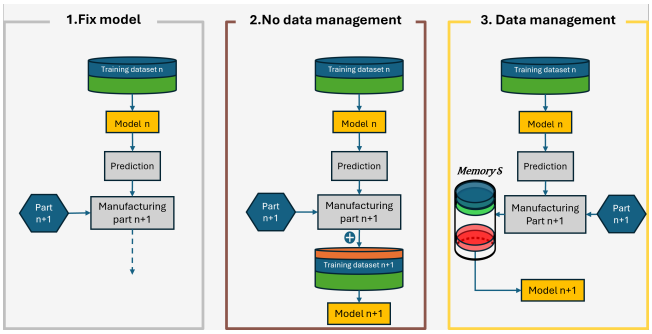


FIGURE 22. **Overview of the three scenarios**

movement. On the CMX 600V, there is an additional complexity layer due to the changed machine with its kinematics. Three scenarios were compared for the validation tests, as outlined in the figure 22.

The components under consideration are run sequentially through the individual scenarios, replicating the sequence of a real production process. Scenario 1 is the baseline approach, where the model is only trained once using an initial training component. No further adjustments or updates are made to the model parameters during the ongoing process. In Scenario 2, all newly available data is added to the training dataset and the model is retrained after each new part. As there is no data management (i.e., no balancing or data reduction) the dataset grows with each new part. The third scenario uses the developed approach. As in scenario 2, the model is retrained after each new part, but the training dataset is managed via the memory $\mathcal{S}$.

B. EXPERIMENT DESCRIPTION

To investigate a progression similar to that in a real production environment, the incremental retrained models are examined one after the other. The tests began with a randomly selected impeller part, the production data of which was used to train the initial ML model. Subsequently, data from the remaining impeller parts are added to the training data in a random order. After each step, the ML model was retrained and its performance evaluated. A random impeller part was selected as the test dataset, excluded from the training data. Thus, one run included 11 iterations, during which the ML model was trained. A total of 30 runs were performed to obtain a statistically reliable result. To ensure the results were independent of the order of the impeller parts and the choice of test part, the order of the impeller parts was randomly generated for each run, and a test part was randomly selected at the beginning of each run. This procedure was used for scenarios 2 and 3. As no retraining took place in scenario 1, the first component in the random list is used for training. This step is repeated 11 times instead of once, so training took place 11 times on a single part, followed by inference on the test part. The same process was followed for the CMX 600V, but with 10 iterations.

C. RESULTS

The DMC results are shown in table 4 and the CMX results in table 5. Both list mean, std, median and number of traning data points N for each scenario. Furthermore, the results are visualized in figure 23 and 24. The statistical significances and effect sizes are shown in table 6. A Shapiro–Wilk test is used to check the distributions on normality, which not all of them are. Therefore, the non-parametric Mann-Whitney U test with a significance level of $\alpha = 0.05$ is used. Additionally, the effect size is determined via Rosenthal's r. [30]

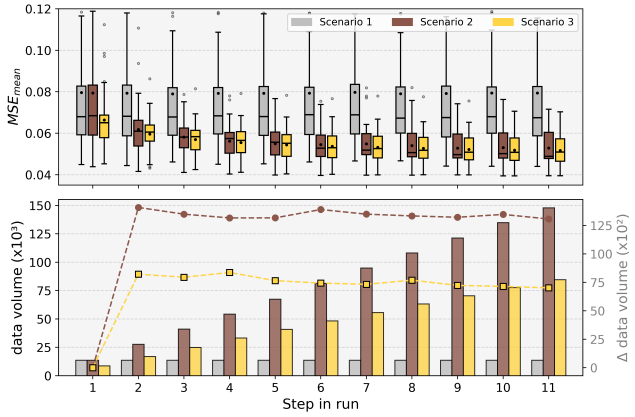


FIGURE 23. Trend for the DMC 60H impeller series

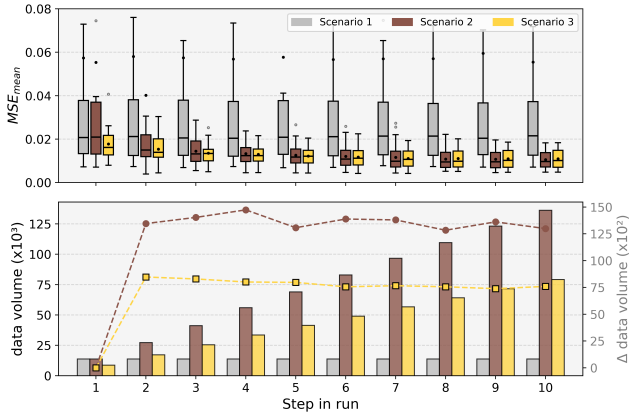


FIGURE 24. Trend for the CMX 600V impeller series

A clear difference emerges between the three scenarios over the 11 iterations. While no significant difference could be detected between scenarios 2 and 3, differences between scenarios 1 and 3, and 2 and 1 become apparent after just a few iterations. This trend is particularly evident on the DMC 60H in figure 23. Although the performance of the model is similar for scenarios 2 and 3, the amount of data required differs significantly. Scenario 3 requires significantly less data, with 57.2% of the data from scenario 2 required after the 11th iteration. The growth rate also shows a downward trend for scenario 3, whereas it remains constant for scenario 2. This is due to memory $\mathcal{S}$ reaching maximum density in an increasing number of bins as the number of iterations increases, meaning that it no longer accepts any further data in well-represented areas.

A similar trend is observed in the CMX 600V results, which supports the general nature of the effect. As can be seen in figure 24, the curves resemble the previously observed pattern. Similarly, Scenarios 2 and 3 show no significant difference but a significantly reduced MSE compared with Scenario 1. Again, the approach presented can be used to significantly reduce the amount of data required. While Scenario 2 required 136 243 data points by the 10th iteration, Scenario 3 required 79 227, corresponding to a reduction of 41.8%.

VII. DISCUSSION

The results demonstrate the potential of the proposed density-controlled replay memory for data balancing and reduction. Figure 25 summarizes the key findings. It shows the curves for the three scenarios, as well as the associated data volume curves. The data volume increases linearly for scenario 2 and remains constant for scenario 1. As scenario 1 does not include any new data, the data volume remains unchanged throughout the iterations. For scenario 3, there is a monotonically increasing curve between scenario 1 and 2. It can also be seen that the curve continues to flatten. Although the number of iterations is insufficient to determine the limit to which the data volume in scenario 3 converges, the figure on the right-hand side indicates convergence as the number of iterations increases. Here, dataset change of scenario 2 represents 100%. Increases in the other iterations are given relative to this growth rate. The difference in the retained datasets between scenarios 2 and 3 continues to increase with each iteration, which supports the hypothesis that the replay memory approaches a bounded data volume.

For scenario 1, a disparity between the mean and median MSE can be seen. The same effect is evident for scenario 2, particularly in the first iteration. Examining all 30 runs of scenario 1 reveals outliers in some of the train and test configurations (see figure 27). This effect represents a dataset-specific perspective on forgetting. Iterative training will result in reduced robustness and fluctuations in model performance. This effect cannot be attributed to a single component. The training dataset in Run 28 is impeller 12, and the model is tested on impeller 8 (see [29]). However, this part is also used for training in run 12 and tested on Impeller 9, resulting in good prediction quality. Therefore, it should be noted that certain unfavorable combinations can lead to poor prediction quality. Nevertheless, this can be compensated for within a few iterations (see scenario 2).

This effect is not apparent in Scenario 3 due to the filtering effect of the data space. This can be seen in the course of the spindle signal in run 28 (see figure 28). In this case, implausible values that do not represent the process are excluded. Therefore, it can be concluded that even a small amount of data can negatively impact prediction quality in unfavorable training-test-configurations. This can be addressed by

TABLE 4. Results for the DMC 60H impeller series. The mean, std, median and number of data points (N) are listed for scenarios 1–3.

Step	Scenario 1				Scenario 2				Scenario 3			
	Mean	std	Median	N	Mean	std	Median	N	Mean	std	Median	N
1	0.07958	0.03967	0.06795	13586	0.07938	0.03951	0.06843	13586	0.06641	0.01556	0.06504	8679
2	0.07934	0.03806	0.06815	13586	0.06178	0.01418	0.06095	27663	0.05957	0.00964	0.06052	16895
3	0.07897	0.03970	0.06784	13586	0.05814	0.00912	0.05811	41148	0.05699	0.00878	0.05834	24838
4	0.07924	0.03917	0.06838	13586	0.05622	0.00933	0.05743	54305	0.05549	0.00846	0.05650	33201
5	0.07926	0.03956	0.06804	13586	0.05497	0.00854	0.05565	67466	0.05439	0.00825	0.05516	40848
6	0.07933	0.03870	0.06893	13586	0.05452	0.00894	0.05281	81358	0.05370	0.00816	0.05303	48278
7	0.07973	0.03890	0.06889	13586	0.05489	0.01042	0.05180	94845	0.05320	0.00827	0.05256	55602
8	0.07902	0.03906	0.06730	13586	0.05407	0.01027	0.05067	108176	0.05271	0.00814	0.05176	63288
9	0.07919	0.03965	0.06757	13586	0.05280	0.00884	0.04964	121390	0.05218	0.00802	0.05085	70514
10	0.07943	0.04001	0.06798	13586	0.05310	0.00953	0.05015	134853	0.05180	0.00757	0.05077	77654
11	0.07940	0.04018	0.06748	13586	0.05286	0.00935	0.04887	147915	0.05166	0.00753	0.05095	84666

TABLE 5. Results for the CMX 600V impeller series. The mean, std, median and number of data points (N) are listed for scenarios 1–3.

Step	Scenario 1				Scenario 2				Scenario 3			
	MW	SD	Median	N	MW	SD	Median	N	MW	SD	Median	N
1	0.05738	0.08740	0.02069	13793	0.05530	0.08370	0.02091	13793	0.01775	0.00684	0.01618	8731
2	0.05806	0.08819	0.02117	13793	0.04018	0.09233	0.01496	27256	0.01537	0.00655	0.01387	17203
3	0.05745	0.08841	0.02051	13793	0.01442	0.00606	0.01314	41292	0.01352	0.00506	0.01339	25494
4	0.05684	0.08627	0.02035	13793	0.01330	0.00494	0.01239	56031	0.01297	0.00461	0.01237	33502
5	0.05771	0.08970	0.02083	13793	0.01257	0.00496	0.01169	69108	0.01218	0.00424	0.01209	41471
6	0.05664	0.08856	0.02109	13793	0.01202	0.00514	0.01080	82991	0.01172	0.00452	0.01121	49030
7	0.05700	0.08441	0.02136	13793	0.01159	0.00568	0.00977	96792	0.01102	0.00422	0.01064	56689
8	0.05707	0.08732	0.02137	13793	0.01077	0.00465	0.00946	109631	0.01102	0.00452	0.00982	64248
9	0.05947	0.09251	0.02035	13793	0.01071	0.00430	0.00951	123251	0.01093	0.00427	0.01018	71632
10	0.05543	0.08506	0.02151	13793	0.01056	0.00385	0.00968	136243	0.01095	0.00421	0.01010	79227

TABLE 6. Results of the Mann-Whitney U test with significance ($\alpha = 0.05$) and Rosenthals r between scenario S1, S2 and S3 for both machines.

Step	DMC 60H						CMX 600V					
	p(S1–S2)	r(S1–S2)	p(S1–S3)	r(S1–S3)	p(S2–S3)	r(S2–S3)	p(S1–S2)	r(S1–S2)	p(S1–S3)	r(S1–S3)	p(S2–S3)	r(S2–S3)
1	0.936 ✗	0.011	0.187 ✗	0.172	0.187 ✗	0.172	0.889 ✗	0.019	0.182 ✗	0.174	0.26 ✗	0.147
2	0.01 ✓	0.33	0.003 ✓	0.384	0.878 ✗	0.021	0.112 ✗	0.206	0.018 ✓	0.305	0.504 ✗	0.088
3	0.001 ✓	0.418	0.0 ✓	0.46	0.623 ✗	0.065	0.005 ✓	0.357	0.002 ✓	0.395	0.82 ✗	0.031
4	0.0 ✓	0.487	0.0 ✓	0.521	0.912 ✗	0.015	0.002 ✓	0.395	0.001 ✓	0.41	0.854 ✗	0.025
5	0.0 ✓	0.519	0.0 ✓	0.546	0.697 ✗	0.052	0.0 ✓	0.462	0.0 ✓	0.487	0.924 ✗	0.013
6	0.0 ✓	0.531	0.0 ✓	0.565	0.854 ✗	0.025	0.0 ✓	0.481	0.0 ✓	0.496	0.912 ✗	0.015
7	0.0 ✓	0.521	0.0 ✓	0.575	0.708 ✗	0.05	0.0 ✓	0.536	0.0 ✓	0.559	0.982 ✗	0.004
8	0.0 ✓	0.527	0.0 ✓	0.592	0.889 ✗	0.019	0.0 ✓	0.578	0.0 ✓	0.542	0.752 ✗	0.042
9	0.0 ✓	0.557	0.0 ✓	0.588	0.889 ✗	0.019	0.0 ✓	0.569	0.0 ✓	0.536	0.741 ✗	0.044
10	0.0 ✓	0.538	0.0 ✓	0.607	0.809 ✗	0.032	0.0 ✓	0.575	0.0 ✓	0.544	0.797 ✗	0.034
11	0.0 ✓	0.534	0.0 ✓	0.607	0.832 ✗	0.029						

increasing the retained training data volume or by suitable data cleaning under the data-centric paradigm.

An axis-specific evaluation can provide additional insight (see figure 26). The direct comparison shows no relevant differences between scenarios 2 and 3 for axes X, Y and Z. However, a notable trend emerges for the main spindle. Overall, the quality of the predictions is high, but scenario 3 generally appears to be superior to scenario 2, providing more consistent results. As both scenarios are inferred on the same data, this difference is due to the filtering effect of the data space. Points not included in scenario 3's training set may be responsible for the distorted inference in Scenario 2, which further highlights the importance of a data-centric approach. This is also confirmed by the comparison of specific predictions (see figure 29).

Figure 30 shows the data volumes as a function of MSE_{mean} for scenarios 1 to 3. Due to the shift in the curve towards the lower left corner, scenario 3 is able to achieve a higher prediction quality with the same amount of data. In other words, scenario 3 requires less data to achieve the same level

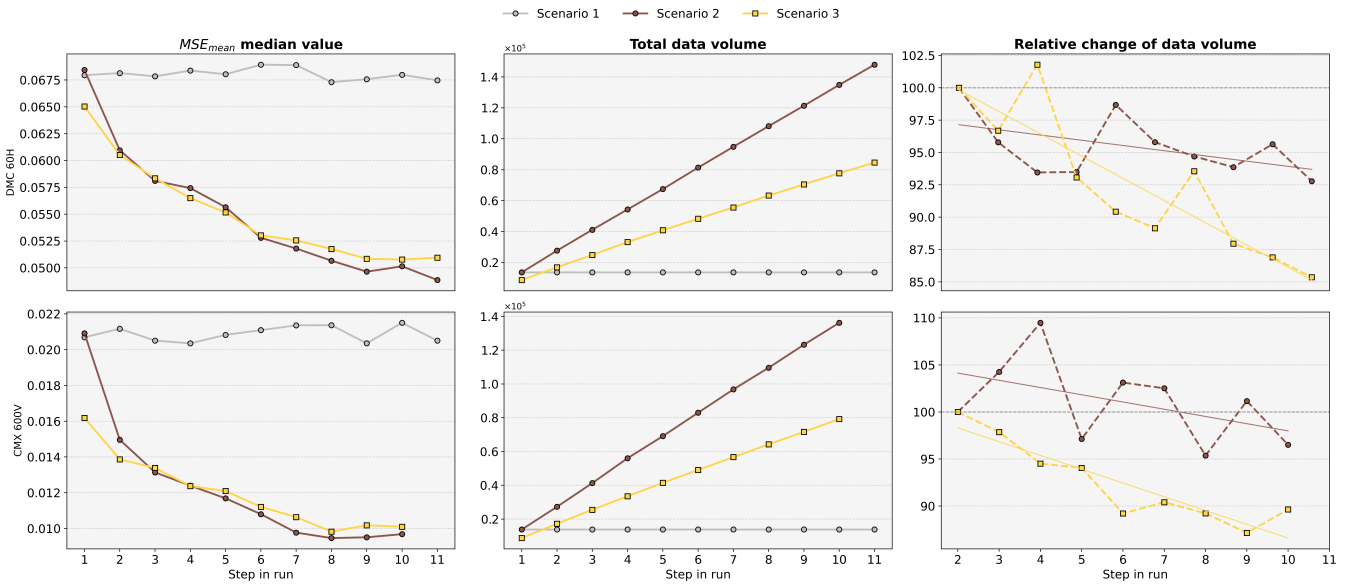


FIGURE 25. Comparison of all scenarios for both machines

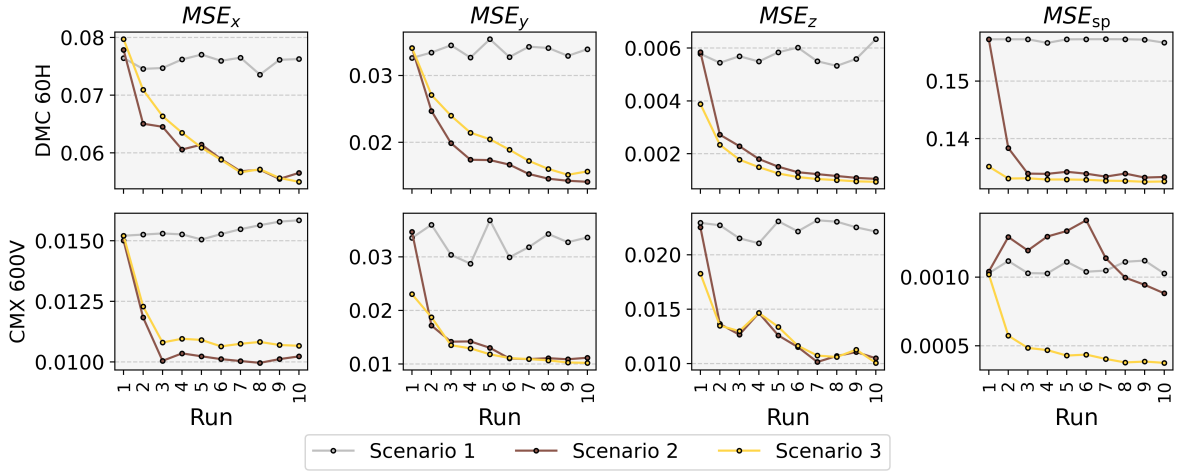


FIGURE 26. Comparison of median MSE_{mean} values of the individual axes

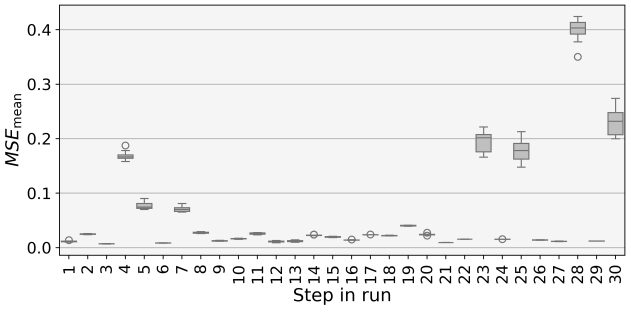


FIGURE 27. Results over all 30 runs of scenario 1 of CMX 600V

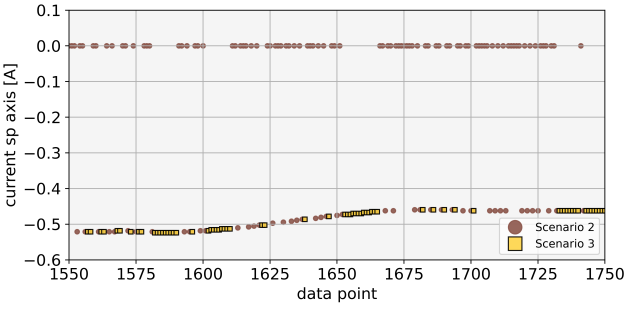


FIGURE 28. Spindle signal of the training data for scenario 2 and scenario 3 for impeller 8

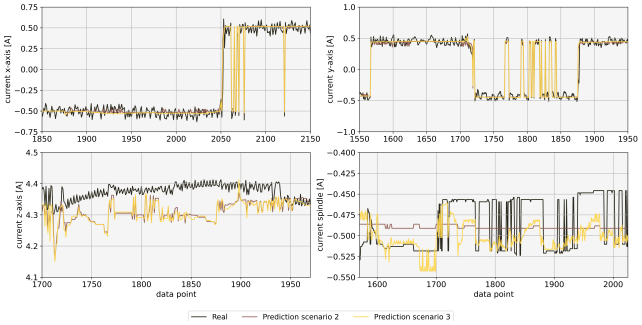


FIGURE 29. Prediction quality for scenario 2 and 3 in direct comparison

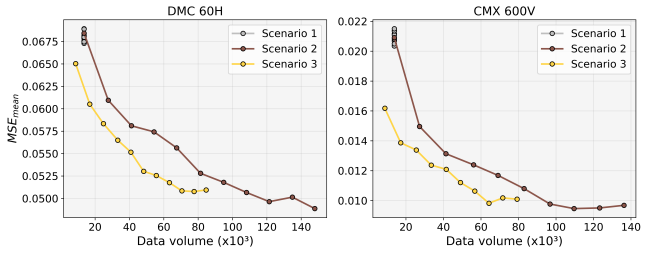


FIGURE 30. Median MSE_{total} as a function of data volume

of prediction quality. Additionally, a convergence trend in the prediction quality can be seen, which is clearer for the second machine. If a lack of information occurs in later iterations despite filled bins, oversampling areas with poor prediction quality may be beneficial.

The effect of targeted data reduction becomes even clearer when the data volume in specific areas is considered. Therefore, selected bins in *AR*-dimension with index values 5 and 6 were considered, corresponding to an *AR* of 5*3.5 to (6+1)*3.5 (*AR*=[17.5,...,24.5]). The data volume for indices $I_{(\mathrm{AR}=5,6),v_c,\varphi}$ can be seen figure 31 and 32. As can be seen, the data volume for scenario 2 shows an almost linear progression, whereas the data volume for scenario 3 decreases. Therefore, the data volume of scenario 3 tends towards a defined amount of data per bin, which restricts the maximum amount of data stored in the replay memory. This has a direct impact on memory usage and the retraining effort.

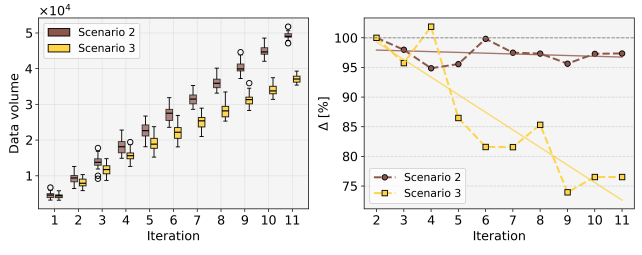


FIGURE 31. DMC 60H data volume in AR={5,6}

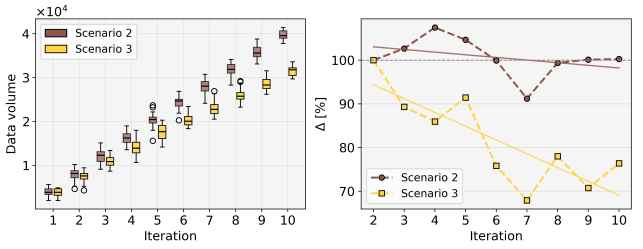


FIGURE 32. CMX 600V data volume in AR={5,6}

A. COMPARISON WITH BASELINE SELECTION STRATEGIES

Data points are selected or rejected based on the density of a given bin. The influence is demonstrated in comparison with Scenario 1 and 2. Since data reduction is controlled by memory $\mathcal{S}$, the data volume after several iterations depends mainly on the distribution of the data points in $\mathcal{S}$ and cannot be predicted in advance. Given the amount of data selected, the approach can be compared with other sampling strategies. Therefore, the performance of Scenario 2 and 3 is compared with random sub-sampling to the reduced amount of Scenario 3 (see table 4 and 5). Furthermore, reservoir sampling is included as a fixed-budget baseline [31]. The reservoir size is set according to the volume of retained data for Scenario 1 in the respective experiment. Since differences are most pronounced in the initial iterations, the comparison focuses on the first six iterations. The results are shown in figure 33 and can also be examined in more detail in the appendix (see table 9 and 10)

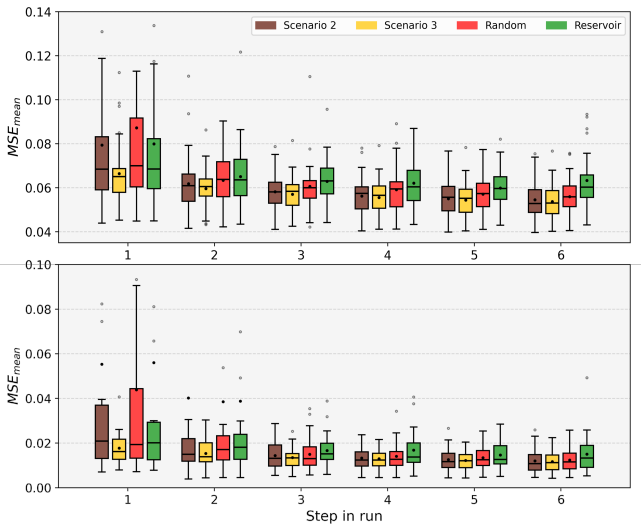


FIGURE 33. Comparison of Scenario 2 and 3 with random sub-sampling and reservoir sampling for the DMC (top) and the CMX (bottom)

The prediction quality for random sub-sampling deteriorates drastically and Scenario 3 is superior up to the third iteration. In later iterations random sub-sampling falls slight behind the other scenarios. It is particularly evident in the early iterations that immediate added value can only be gener-

ated by a targeted selection of suitable data points. For larger datasets, random sub-sampling can also reduce the required data volume efficiently. However, it should be noted that the target data volume is not known in advance. Only a targeted data management, as in Scenario 3, can ensure that no relevant data is excluded. However, if the associated experimental effort is not feasible, this approach could be used to identify an appropriate amount of data for random sub-sampling, which can then be transferred to other problems.

A comparison with reservoir sampling shows that a fixed-size random retention strategy does not offer the same advantages as the proposed density-controlled replay memory. Reservoir sampling yields relatively stable results over iterations, but with limited improvement. This is expected, given that the fixed reservoir size restricts the volume of retained data and does not account for the data distribution. Increasing the size of the reservoir would likely result in behavior approaching that of random sub-sampling. In contrast, Scenario 3 selects data based on bin density and therefore preserves a more representative coverage in $\mathcal{M}$. Consequently, Scenario 3 achieves better prediction quality than reservoir sampling, indicating that the proposed method is superior to a purely random fixed-memory strategy.

The findings support the data-centric AI paradigm, which emphasize the importance of improving data quality in order to enhance model performance. In incremental data generation settings, large and often imbalanced data sets become available over time. Therefore, efficient data reduction and balancing strategies are essential to limit the growth of the retained data while preserving representative coverage of the data space. The baseline comparison shows that the benefit of the density-controlled replay memory is not solely caused by reducing the volume of retained data. The improvement results from the structured selection of representative data points in $\mathcal{M}$. This prevents frequently repeated process regions form dominating the memory while still allowing adaptation to newly emerging operating conditions. Since the retained data volume directly affects storage requirements and retraining effort, the next section discusses the practical system-level implications of the proposed replay memory.

B. PRACTICAL SYSTEM-LEVEL CONSIDERATIONS

The proposed approach is motivated by the need to limit the continuous growth of training datasets in incremental retraining scenarios. From a system-level perspective, reducing the volume of retained data is directly relevant to memory usage, data handling and the effort involved in retraining. Unlike full incremental data expansion in scenario 2, the density-controlled replay memory of scenario 3 limits the number of stored samples by accepting or rejecting data points based on the density of their corresponding bin. Consequently, the replay memory prevents unlimited growth in densely populated regions of the data space. This reduction in retained data volume directly reduces the storage required for historical training data. Since subsequent retraining is performed on the retained dataset, a smaller replay memory reduces the amount of data that must be loaded, processed and used during these iterations. Therefore, the reported reduction in data volume can be interpreted as an indicator of reduced memory usage and lower retraining effort.

The computational overhead of updating the memory is limited to assigning incoming data points to the reduced feature representation and checking the density of the corresponding bin. This operation is lightweight compared with the computational cost of model retraining, since it does not involve optimizing the predictive model's parameters. The main computational benefit is therefore expected during retraining, where the reduced replay memory limits the number of samples used for model updates while maintaining representative coverage of the data distribution. However, a full runtime analysis is not included in the present evaluation. The time taken for retraining, the memory used for implementation and the computational cost depend on the hardware, the software implementation, the model type, the batch size and the retraining schedule. Therefore, this study primarily evaluates efficiency through retained data volume and prediction quality. A detailed, deployment-oriented runtime analysis is an important aspect of future work.

C. LIMITATIONS AND GENERALITY

The results demonstrate the effectiveness of the proposed density-controlled replay memory in the CNC milling scenarios investigated. The generality of the proposed feature-space representation is currently restricted by the process conditions considered in this study. All parts evaluated are produced in planar milling operations, resulting in $v_z \approx 0$ during machining. While the present validation supports the applicability of the approach to planar milling processes, it does not yet demonstrate its transferability to machining operations involving significant Z-axis motion or complex, three-dimensional tool paths. Relately, the reduced feature representation $\mathcal{M} = (v_c, \varphi, AR)$ is specifically designed for the investigated planar setting. For milling processes involving substantial Z-axis engagement or 5-Axis milling, an extended representation can be beneficial to adequately capture the relevant process states. Therefore, the proposed feature representation should not be interpreted as applicable to all CNC milling operations, but rather as an effective and physically interpretable representation of the evaluated planar regime.

The parameter values for the replay memory $\mathcal{S}$ are determined based on aluminum and steel. The validation on POM-C indicates transferability across the investigated materials and tools. However, different machines, tools, materials or machining strategies may necessitate an adaptation. While the overall approach is model-agnostic and does not rely on a particular model architecture, this study uses ExtraTrees and Gradient Boosting models. Other model types or regression tasks may exhibit different sensitivities and require task-specific parameter tuning. These limitations define the current scope of applicability and emphasize key areas for future research.

VIII. SUMMARY AND OUTLOOK

This paper presents a data-centric approach for dataset balancing and management in incremental retraining scenarios, exemplified by its application to the prediction of axis-specific current signals of three-axis milling machines. It is based on a dataset consisting of several parts manufactured using a DMC 60H three-axis milling machine from DMG Mori. Initially, a reduced feature representation $\mathcal{M}$ was introduced to represent the data in a compact form interpretable by a human user. Subsequently, this feature representation was discretized to construct a memory $\mathcal{S}$, using the parameters $v_{\mathrm{c\,max}}$, $\mathrm{AR}_{\mathrm{max}}$, V_{Bin}, Δv_c and $\Delta\mathrm{AR}$. This memory was then used to control and balance the training data distribution over time via a target density ρ_{max}. To determine the initial parameter values, the distribution of the development data for the two materials (aluminum and steel) was considered, and specific parameter values were derived. Subsequent tests optimized and adjusted these initial values to material-independent parameter values. Additionally, an oversampling study was conducted to investigate its effects in specific areas of the memory $\mathcal{S}$.

The memory $\mathcal{S}$, is validated and analyzed in a series of individual parts on two machines with new tool paths and technological parameters. Overall, the developed concept can be confirmed, revealing clear advantages in terms of the required amount of data. While the show no significant deterioration in performance, the required data volume could be reduced by over 40% across the iterations. Furthermore, the filtering effect of the data space emphasized the importance of data cleaning under the data-centric AI paradigm. These results indicate that a density-controlled replay memory enables effective incremental retraining by managing the training data distribution, providing a complementary solution to model-centric approaches.

Future investigations should focus on deploying the proposed approach in long-term, real-world process monitoring scenarios. Of particular interest is the behavior of the density-controlled replay memory with respect to data convergence, retained data volume and retraining effort over extended operational periods. As the present study focuses on developing and validating the data balancing strategy, a detailed evaluation of runtime, memory consumption and retraining time under deployment conditions was not performed. These system-level characteristics depend on specific hardware and software implementations, retraining schedules, and model configurations, and therefore remain an important subject for future work. In this context, a broader set of industrially relevant performance metrics, as well as the interaction between the data-centric replay memory and a model centric hyperparameter optimization, should be investigated.

Future work should also explore the integration of the proposed replay mechanism with AL strategies. Although, the oversampling interface was demonstrated through targeted experiments, a fully automated incremental decision system remains future work. In particular, formal trigger criteria and decision logic for when and where to oversample the replay memory, as well as the AL strategy for selecting additional data points, should be investigated. Furthermore, using augmented or synthetic data for targeted enrichment of specific regions in the feature representation offers additional potential for dataset optimization.

ACKNOWLEDGMENT

This Project was supported by the Federal Ministry for Economic Affairs and Climate Action (BMWK), based on a decision by the German Bundestag via Gesellschaft zur Förderung angewandter Informatik e.V.—GFaI. (Grant No. 22849 BG/2). The KIT-Publication Fund of Karlsruhe Institute of Technology provided support for this publication.

References

[1] M. Bortolini, F. G. Galizia, and C. Mora, "Reconfigurable manufacturing systems: Literature review and research trend," *Journal of Manufacturing Systems*, vol. 49, pp. 93–106, Oct. 2018. DOI: 10.1016/j.jmsy.2018.09.005.

[2] R. Teti, D. Mourtzis, D. D'Addona, and A. Caggiano, "Process monitoring of machining," *CIRP Annals*, vol. 71, no. 2, pp. 529–552, 2022. DOI: 10.1016/j.cirp.2022.05.009.

[3] R. Ströbel et al., "Hybrid machine learning for cnc process monitoring," *IEEE Access*, vol. 13, pp. 91 875–91 888, 2025. DOI: 10.1109/ACCESS.2025.3573400.

[4] S. Wares, J. Isaacs, and E. Elyan, "Data stream mining: Methods and challenges for handling concept drift," *SN Applied Sciences*, vol. 1, no. 11, p. 1412, Nov. 2019. DOI: 10.1007/s42452-019-1433-0.

[5] Y. Y. Bay and K. A. Yearick, *Machine learning vs deep learning: The generalization problem*, 2024. [Online]. Available: https://arxiv.org/abs/2403.01621.

[6] A. Verma, A. Goyal, and S. Kumara, "Machine learning-assisted collection of reduced sensor data for improved analytics pipeline," *Procedia CIRP*, vol. 121, pp. 150–155, 2024. DOI: 10.1016/j.procir.2023.09.242.

[7] A.-J. Gallego, J. Calvo-Zaragoza, and R. B. Fisher, "Incremental unsupervised domain-adversarial training of neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 11, pp. 4864–4878, Nov. 2021. DOI: 10.1109/TNNLS.2020.3025954.

[8] G. M. v. d. Ven, N. Soures, and D. Kudithipudi, "Continual learning and catastrophic forgetting," in *Learning and Memory: A Comprehensive Reference*, Third Edition, vol. 1, Academic Press, Oxford, 2025, pp. 153–168. DOI: 10.1016/B978-0-443-15754-7.00073-0.

[9] G. M. Van De Ven, T. Tuytelaars, and A. S. Tolias, "Three types of incremental learning," *Nature Machine Intelligence*, vol. 4, no. 12, pp. 1185–1197, Dec. 5, 2022. DOI: 10.1038/s42256-022-00568-3.

[10] J. T. Vogelstein et al., "Simple lifelong learning machines," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 47, no. 11, pp. 10 033–10 046, Nov. 2025. DOI: 10 . 1109 / TPAMI . 2025 . 3595364.

[11] M. Jehanzeb Mirza, M. Masana, H. Possegger, and H. Bischof, "An efficient domain-incremental learning approach to drive in all weather conditions," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, New Orleans, LA, USA: IEEE, Jun. 2022, pp. 3000–3010. DOI: 10.1109/CVPRW56347.2022.00339.

[12] M. Wulfmeier, A. Bewley, and I. Posner, "Incremental adversarial domain adaptation for continually changing environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 1–9. DOI: 10.1109/ICRA.2018.8460982.

[13] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, and K. Alahari, "End-to-end incremental learning," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., vol. 11216, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2018, pp. 241–257. DOI: 10.1007/978-3-030-01258-8_15.

[14] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "Icarl: Incremental classifier and representation learning," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, Jul. 2017, pp. 5533–5542. DOI: 10.1109/CVPR.2017.587.

[15] J. Jakubik, M. Vössing, N. Kühl, J. Walk, and G. Satzger, "Data-centric artificial intelligence," *Business Information Systems Engineering*, pp. 1–9, Mar. 2024. DOI: 10.1007/s12599-024-00857-8.

[16] S. E. Whang, Y. Roh, H. Song, and J.-G. Lee, "Data collection and quality challenges in deep learning: A data-centric AI perspective," *The VLDB Journal*, vol. 32, no. 4, pp. 791–813, Jul. 2023. DOI: 10.1007/s00778-022-00775-9.

[17] N. Polyzotis and M. Zaharia, *What can data-centric AI learn from data and ML engineering?* Dec. 13, 2021. DOI: 10.48550/arXiv.2112.06439.

[18] R. Ströbel, M. Mau, A. Puchta, and J. Fleischer, "Improving time series regression model accuracy via systematic training dataset augmentation and sampling," *Machine Learning and Knowledge Extraction*, vol. 6, no. 2, pp. 1072–1086, May 11, 2024. DOI: 10.3390/make6020049.

[19] L. Camacho, G. Douzas, and F. Bacao, "Geometric smote for regression," *Expert Systems with Applications*, vol. 193, p. 116 387, 2022. DOI: 10.1016/j.eswa.2021.116387.

[20] P. Kumar and A. Gupta, "Active learning query strategies for classification, regression, and clustering: A survey," *Journal of Computer Science and Technology*, vol. 35, no. 4, pp. 913–945, Jul. 2020. DOI: 10.1007/s11390-020-9487-4.

[21] J. Fonseca and F. Bacao, "Improving active learning performance through the use of data augmentation," *International Journal of Intelligent Systems*, vol. 2023, no. 1, F. E. Petry, Ed., p. 7 941 878, Jan. 2023. DOI: 10.1155/2023/7941878.

[22] D. Cacciarelli and M. Kulahci, "Active learning for data streams: A survey," *Machine Learning*, vol. 113, no. 1, pp. 185–239, Jan. 2024. DOI: 10.1007/s10994-023-06454-2.

[23] N. Bhatt, N. Bhatt, P. Prajapati, V. Sorathiya, S. Alshathri, and W. El-Shafai, "A data-centric approach to improve performance of deep learning models," *Scientific Reports*, vol. 14, no. 1, p. 22 329, Sep. 27, 2024. DOI: 10.1038/s41598-024-73643-x.

[24] X. Hu et al., "PSRONet: A Deep Reinforcement Learning-Based Sensor Configuration Framework in Railway Point Machines Fault Diagnosis," en, *IEEE Transactions on Instrumentation and Measurement*, vol. 75, pp. 1–13, 2026. DOI: 10 . 1109 / TIM . 2026 . 3654732. Accessed: Apr. 7, 2026.

[25] A. S. Raihan et al., "A data-efficient sequential learning framework for melt pool defect classification in laser powder bed fusion," *Journal of Manufacturing Processes*, vol. 145, pp. 201–210, Jul. 2025, ISSN: 15266125. DOI: 10.1016/j.jmapro.2025.03.118.

[26] R. Ströbel et al. "Part series dataset of milling processes for time series prediction," Accessed: Oct. 20, 2025.

[27] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19, Anchorage, AK, USA: Association for Computing Machinery, 2019, pp. 2623–2631. DOI: 10.1145/3292500.3330701.

[28] R. Ströbel et al. "A Multimodal Dataset for Process Monitoring and Anomaly Detection in Industrial CNC Milling. "[Online]. Available: https://publikationen.bibliothek.kit.edu/1000182633.

[29] R. Ströbel, M. Kuck, F. Oexle, A. Puchta, and J. Fleischer. "A Multimodal Dataset 2 for Process Monitoring and Anomaly Detection in Industrial CNC Milling. "[Online]. Available: https://publikationen.bibliothek.kit.edu/1000185172.

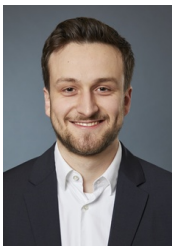
[30] A. Field, *Discovering statistics using SPSS: and sex and drugs and rock'n'roll*, en, 3. ed., repr. Los Angeles: Sage, 2012.

[31] J. S. Vitter, "Random sampling with a reservoir," en, *ACM Transactions on Mathematical Software*, vol. 11, no. 1, pp. 37–57, Mar. 1985. DOI: 10.1145/3147.3165. Accessed: May 25, 2026.

ROBIN STRÖBEL received his M.Sc. in mechanical engineering at Karlsruhe Institute of Technology (KIT) in 2022. Since 2022, he is a research associate at the Intelligent Machines and Components research group at wbk Institute of Production Science, where he works on various projects in the field of data science in manufacturing. He is currently pursuing a Ph.D. degree in incremental model training for model based anomaly detection.

AARON BÜTTNER received his M.Sc. degree in business management and engineering at Karlsruhe Institute of Technology (KIT) in 2026. His research interests are in the area of production technology, process monitoring and data science.

HAFEZ KADER is a PhD student in the Autonomous Multisensor Systems (AMS) research group at the Faculty of Computer Science at the Otto-von-Guericke University Magdeburg (OVGU). His research focuses on analysing the relevance and significance of different features and detecting anomalies in sensor data to make processes more efficient and optimise the performance of machine learning models.



ALEXANDER PUCHTA received the M.Sc. degree in mechanical engineering from Karlsruhe institute of Technology, in 2020. He is currently pursuing the Ph.D. degree in machine, plant, and process automation, focusing on the topic of autonomous modeling of feed axes for machine tools. He has been the Head of the Intelligent Machines and Components Group, wbk Institute of Production Science, since 2023.

BENJAMIN NOACK studied computer science at the University of Karlsruhe (TH) and received his PhD in 2013 from the Intelligent Sensor-Actuator Systems (ISAS) research group. Since 2021 he is a professor at the Faculty of Computer Science at the Otto-von-Guericke University Magdeburg (OVGU), where he heads the Autonomous Multisensor Systems (AMS) research group. His research focuses on distributed sensor data fusion and its applications in industrial process monitoring and autonomous mobile robotics.

JÜRGEN FLEISCHER received the degree in mechanical engineering from Universität Karlsruhe (TH), and the Ph.D. degree from the Institute for Machine Tools and Industrial Engineering (iwb), in 1989. Since 1992, he has been holding several leading positions in industry before being appointed as a Professor and the Head of the wbk Institute for Production Technology, Karlsruhe Institute of Technology (KIT), in 2003. In addition, he has been a Visiting Professor with Tongji University, Shanghai, since 2012. His current scientific research focuses are intelligent components for machine tools and handling systems, the automation of immature processes, and agile production facilities. As a recognized member of the scientific community, he is active in German Academy of Science and Engineering (acatech), and is a member of several scientific and industrial advisory boards.

IX. APPENDIX

Figure 34 and 35 show the complementary plots for Experiment 2. Table 7 and 8 present the updated search space and the unified results of Experiment 3. Figure 36-41 present supplementary results for Experiment 4. Furthermore, Table 9 and 10 summarize the validation results for random subsampling and reservoir sampling.

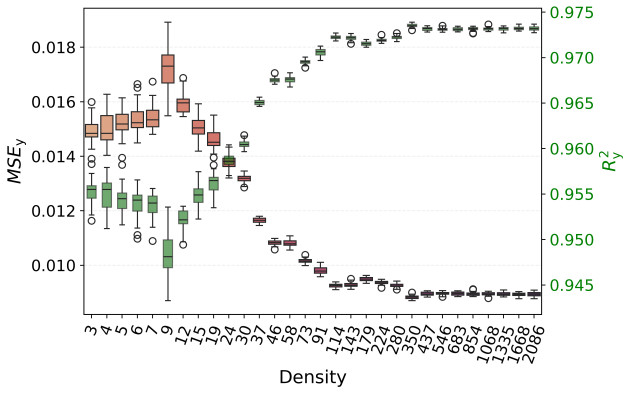


FIGURE 34. MSE and R^2 for the Y-axis (aluminum)

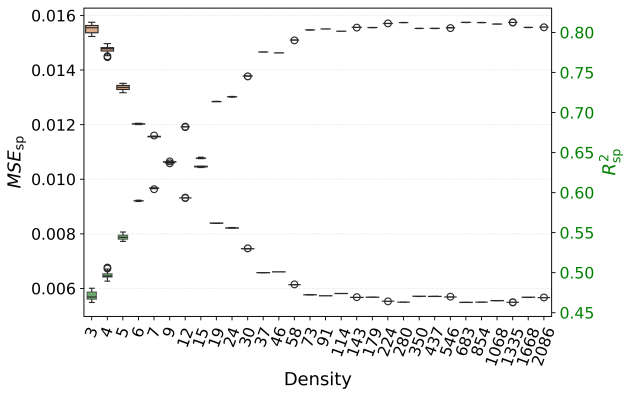


FIGURE 35. MSE and R^2 for the SP-axis (aluminum)

TABLE 7. Updated parameter space

Parameter	Search areas
ρ	91, 114, 143, 179, 193, 224, 280
Δv_c	[0.3,...,0.6] ; step = 0.1
V_{Bin}	[0.35,...,2] ; step=0.02
ΔAR	3.5

TABLE 8. Optimal combinations of parameters for aluminum and steel

position	aluminum Δv_c	aluminum V_{Bin}	aluminum $\rho_{\max}$	aluminum $\mathrm{MSE}_{\mathrm{total}}^{\mathrm{sum}}$	steel Δv_c	steel V_{Bin}	steel $\rho_{\max}$	steel $\mathrm{MSE}_{\mathrm{total}}^{\mathrm{sum}}$
1	0.6	1.25	280	0.116372	0.4	1.85	91	1.437117
2	0.6	1.29	280	0.116770	0.4	0.87	224	1.437311
3	0.5	1.87	280	0.117544	0.4	0.93	224	1.438055
4	0.5	1.43	224	0.117710	0.3	0.87	224	1.438349
5	0.6	1.11	280	0.117782	0.4	1.05	224	1.438451
6	0.6	1.23	280	0.117844	0.4	0.41	224	1.438910
7	0.6	1.85	280	0.118005	0.4	0.83	224	1.439949
8	0.6	1.43	280	0.118025	0.4	0.91	224	1.430199
9	0.5	1.87	280	0.118043	0.4	1.65	224	1.430329
10	0.6	1.83	280	0.118056	0.4	0.87	224	1.430729
mean	$\Delta v_c = 0.5,\ V_{\mathrm{Bin}} = 1.3,\ \rho_{\max} = 240,\ \Delta\mathrm{AR} = 3.5$							
$\mathrm{MSE}_{\mathrm{total}}$	0.1201 (-0.0037)				1.4430 (-0.0059)			

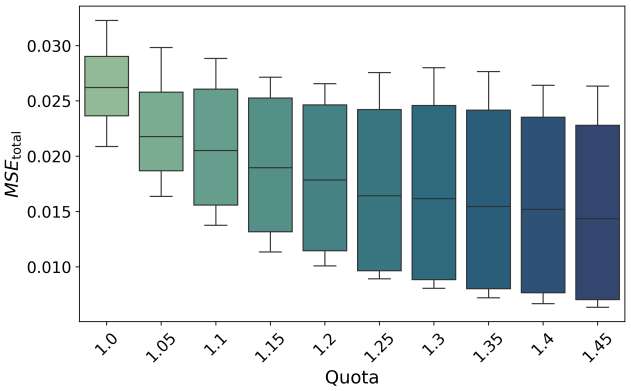


FIGURE 36. $\mathrm{MSE}_{\mathrm{total}}$ for notch bin 1, 2, 3, and 5

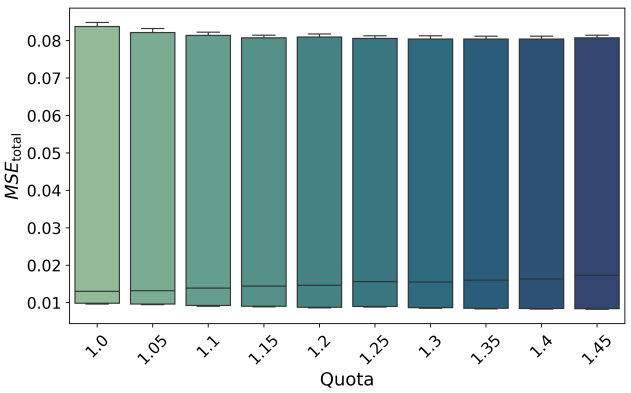


FIGURE 37. $\mathrm{MSE}_{\mathrm{total}}$ of the three plate origin bins 1, 3, and 5

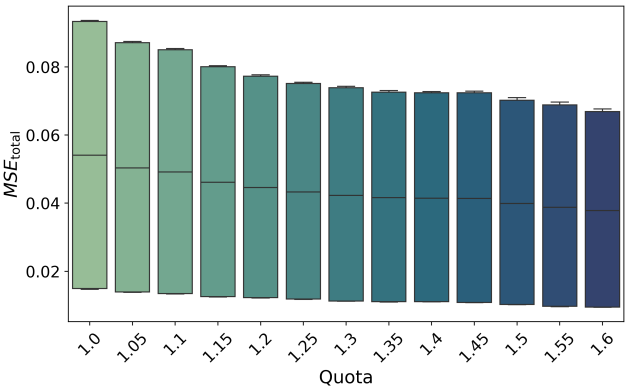


FIGURE 38. $\mathrm{MSE}_{\mathrm{total}}$ for plate bin 3 and 5

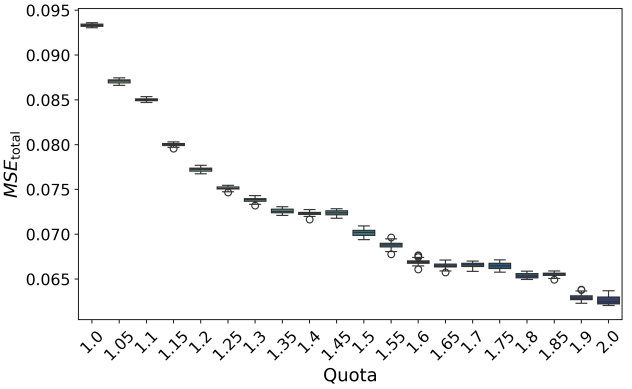


FIGURE 39. $\mathrm{MSE}_{\mathrm{total}}$ for plate bin 5

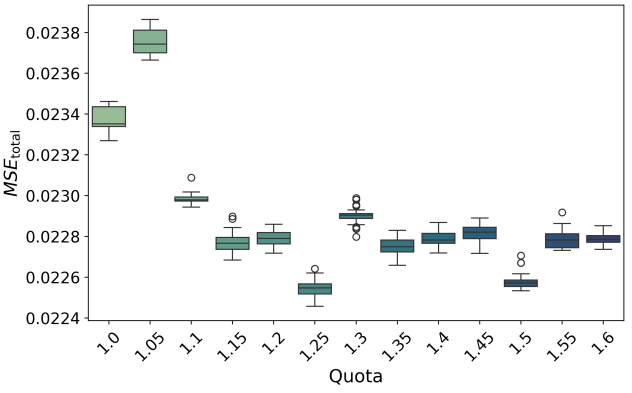


FIGURE 40. $\mathrm{MSE}_{\mathrm{total}}$ of the notch bin 3

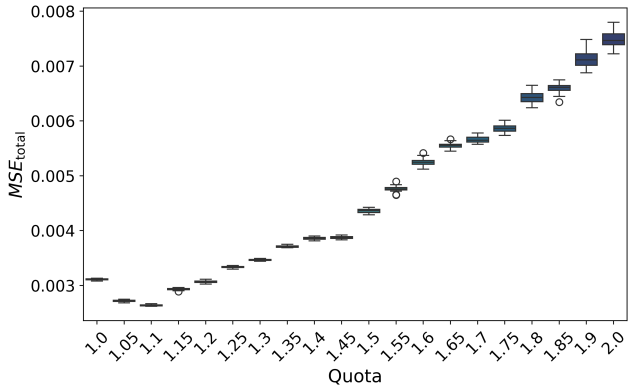


FIGURE 41. MSE_{total} of the notch bin 5

TABLE 9. Results for the DMC 60H impeller series. The mean, std, median and number of data points (N) are listed for random sub-sampling and reservoir sampling.

Step	Scenario 4				Scenario 5			
	MW	SD	Median	N	MW	SD	Median	N
1	0.08722	0.05048	0.06999	8679	0.07989	0.03995	0.06849	13586
2	0.06336	0.01136	0.06380	16895	0.06506	0.01504	0.06358	13586
3	0.06056	0.01208	0.05998	24838	0.06290	0.01114	0.06310	13586
4	0.05909	0.01146	0.05944	33201	0.06208	0.01143	0.06037	13586
5	0.05708	0.00928	0.05737	40848	0.05985	0.00949	0.05964	13586
6	0.05592	0.00897	0.05583	48278	0.06326	0.01277	0.06028	13586

TABLE 10. Results for the CMX 600V impeller series. The mean, std, median and number of data points (N) are listed for random sub-sampling and reservoir sampling.

Step	Scenario 4				Scenario 5			
	MW	SD	Median	N	MW	SD	Median	N
1	0.04388	0.06578	0.01932	8731	0.05604	0.08515	0.02013	13793
2	0.03852	0.07827	0.01709	17203	0.03876	0.07564	0.01812	13793
3	0.01500	0.00735	0.01306	25494	0.01665	0.00694	0.01518	13793
4	0.01407	0.00615	0.01270	33502	0.01684	0.00824	0.01376	13793
5	0.01343	0.00504	0.01255	41471	0.01466	0.00623	0.01263	13793
6	0.01238	0.00505	0.01158	49030	0.01506	0.00852	0.01332	13793