Towards a real-time tool state detection in sheet metal forming processes validated by wear classification during blanking

C Kubik1,*, D A Molitor1, M Rojahn1 and P Groche1

1 Institute for Production Engineering and Forming Machines, Darmstadt, Germany

* kubik@ptu.tu-darmstadt.de

Abstract. The potential of data for inline detection of changes in the physical state of sheet metal forming processes has been proven over the last decade. However, with production rates exceeding 300 parts per minute the time available for a workpiece-related processing of sensor data is reduced. Therefore, the analysis of large data sets is outsourced to the cloud taking advantage of the high computing power provided there. But within this cloud-based computing paradigm, the speed of data transmission hinders real-time analysis of data and causes latency between fault detection and its occurrence. To overcome this bottleneck, this study aims to evaluate a data-based monitoring (DBM) approach for estimating process states in high speed sheet metal forming in terms of their suitability for a decentralized analysis at the edge. Thereby, the DBM is evaluated according to the model accuracy and the absolute computing time. In order to quantify these key performance parameters and the applicability of the DBM on edge devices, a classification of 16 wear states during blanking is considered. Based on the key performance parameters, an optimal DBM approach for decentralized analysis is proposed and an empirical formulation is provided to estimate the absolute computing time depending on the computational resources used for data processing.

1. Introduction

The increasing availability of data enables companies to master challenges of modern manufacturing processes. Therefore, manufacturers aim to use knowledge gained from data to fulfill the increased demands on product quality by ensuring simultaneously an economical production [1]. In this context, process-related uncertainties, such as varying material properties, temperature conditions or wear, must be controlled to ensure this specified product quality. Especially in the manufacturing process of blanking, unexpected wear phenomena are a major reason for insufficient product quality, forcing an increase of scrap. Due to the strong tribological loads resulting from high contact pressure and sliding distances, blanking tools are exposed to abrasive and adhesive wear in the forming zone [2]. Even though wear has a significant impact on the blanking process and the quality of the processed workpieces, determining the state of wear requires additional effort by interrupting the process, removing and manually inspecting the tool [3]. This leads to a conflict between minimizing machine downtimes (manufacturer side) and maximizing product quality (customer side). For this reason, real-time estimation of wear states during blanking is mandatory for scheduling maintenance intervals and maximizing the machine utilization while ensuring a desired product quality.

In this context, DBM represents a successful technique for in-line estimation of process states and can reduce or prevent unplanned machine downtime. A study by McKinsey & Company shows that
DBM reduces machine downtime by 30 to 50 percent and extends the machine life by 20 to 40 percent [4]. Here, DBM aims to automatically extract hidden information from a heterogeneous database and quantify the current process state based on this knowledge. However, manually processing of the huge amount of available data in high speed blanking processes complicates this knowledge discovery procedure and requires advanced analysis techniques such as machine learning (ML) [5] or feature-based supervision (FS) [6]. In the context of in-line wear estimation during blanking, DBM approaches such as ML or FS are able to quantify complex and non-linear interdependencies between wear phenomena and a sensorial acquired variable [7].

Next to the discovery of knowledge from data, data management plays a crucial role, especially in blanking operations, where production rates of over 300 stroke per minute (spm) are achieved [8]. In this process, large amounts of measurement data are generated in a short time, which must be analyzed in relation to the workpiece. For a blanking process at 300 spm, and a material thickness of 0.3 mm, this results in a tool engagement time of less than 334 ms. In this case, the time frame of 334 ms for data acquisition, data processing and adjusting the process must be smaller than the time for one stroke due to the knowledge gained [9].

In practice, the data processing step is outsourced to external computational sources, due to the high computing power provided, hereinafter referred to as cloud. However, compared to the rapidly developing data processing speed and computing power in the cloud, bandwidth improvements are at a standstill. With the increasing amount of data, the data transmission speed becomes a bottleneck in the cloud-based computing paradigm [10]. To ensure a real-time DBM, data must be processed at the origin where it is gathered. A decentralized data processing leads to better bandwidth availability, a faster data transmission and reduced latency. Therefore, it is necessary to shift the procedure of DBM from the cloud to the edge [11,12]. In this study, edge is defined as a device that enables decentralized data processing directly at the source of data generation [13]. In contrast to the cloud, edge devices rely on their local computing capacities. Therefore, it is important to keep the required computing power for edge devices as low as possible. At the same time, the quality of the generated knowledge and thus the reliability of the inline estimated process state should not be limited. Figure 1 compares the paradigm of edge and cloud computing, adapted to a manufacturing process. Thereby, data is provided by the process via sensors and centrally (a) or locally (b) processed. As a result of the analysis, knowledge is provided, that allows the process to be adapted to varying boundary conditions. The quality of the discovered knowledge and, related to this, the identification of the current process state is decisive for an appropriate adaptation of the process.

![Figure 1: Comparison between cloud- and edge-based DBM related to manufacturing processes](image)

Therefore, the objective of this work is to evaluate DBM for real-time estimation of process states in high speed sheet blanking with regard to their suitability for decentralized analysis at the edge. Thereby, the efficiency of DBM is quantified by the model accuracy (performance) and the absolute computing time calculated from the computational effort of the central processing unit (CPU). To quantify the efficiency of the DBM approaches, a classification of sixteen wear states is considered during a blanking operation at production rates up to 300 spm. Based on the two key parameters (model performance and absolute computing time) an optimal DBM for decentralized analysis at the edge is proposed. Therefore, sections 2 and 3 provide an introduction to blanking and the relevance of wear for this forming operation. Furthermore, an overview of existing DBM in sheet metal forming is given. Section 4 presents the experimental data sets with a total of about 4.000 strokes, the two key parameters and the training
procedure for the model. Afterwards in section 5, the efficiency of the DBM approaches for a decentralized estimation of 16 abrasive wear state on the edge is quantified. In order to demonstrate the performance of the classification marginally differing wear conditions states are defined in 16 narrow intervals. Finally, section 6 summarizes the available results and provides a roadmap for future work.

2. Wear phenomena during blanking

As a result of the relative motion of the tool and the sheet metal at high stroke speed combined with high tribological loads, wear significantly affects the conditions of the blanking process. Thereby, the motion of the tool is characterized by three phases as shown in Figure 2 (a). During the punch-phase (I), the tool hits the sheet, causing an abrupt increase in the blanking force. The material changes from a linear elastic phase to a plastic deformation. In the process, the blanking force decreases until the material breaks completely. While in the push-phase (II) the blanked workpiece is ejected from the die, in the withdraw-phase (III) the tool is pulled out of the die. In both phases, the blanking force deviates from the zero level due to jamming between the blanked workpiece and the die and between the punch and the sheet metal strip. In addition to the relative motion, high contact stresses, reduced lubrication, large sliding distances and the processing of high-strength materials accelerate wear [14,15].

During blanking, wear is initiated by surface adhesions in the area of the forming zone. Due to the high contact stresses in the forming zone, the surfaces of the tool and sheet stick together. The relative motion and the jamming between the tool and the sheet cause frictional forces, that tear out small parts of the tool and cause the formation of wear particles. High abrasion is to be expected, especially near the cutting edge of the punch. According to Archard, the wear volume linearly depends on the contact normal stress and causes decreasing wear from the bottom of the punch upwards along the lateral surface [16]. In preliminary studies, a correlation was found between the contact normal stress and the distance from the bottom of the punch, as shown in Figure 2 (b). Thereby, the black line describes the outer contour of the cutting edge of the punch and the blue line the contact normal stresses along these outer contour.

![Sensorial acquired force-displacement curve during blanking process](Image)

**Figure 2:** Blanking tool for acquiring force curves (a) and dependency between simulated contact normal stresses $\sigma_N$ and the distance from the bottom of the punch (b)

As shown in the literature, the onset of wear as well as the type of wear are influenced by the material (sheet metal and tool), tool parameters (clearance and edge radii) [17,18] stroke speed, lubrication, and material properties (sheet thickness and alloy constituents) [19,20]. In particular, the material properties play a decisive role in the type of wear that occurs. According to Hohmann et al., soft graded steels ($R_m < 350$ MPa) tend to adhere to the harder surface of the tool. In contrast, high strength steel grades ($R_m > 600$ MPa) tend to cause abrasive wear on the edge of the blanking tool [21]. Despite this tendency, soft graded steels also show abrasive wear phenomena [22,23]. In this case, excessive wear occurs after a longer sliding distance (number of strokes) due to the lower contact normal pressure of the soft graded steels. Since the trend is towards processing increasingly high-strength materials, abrasive wear is occurring more frequently [24]. This leads to shorter maintenance intervals and a deterioration of the tool geometry in the form of a rounded cutting edge. In addition, the change in the tool geometry correlates with the quality of the blanked parts [3].
3. Data-based prediction of tool wear in sheet metal forming

Real-time DBM is mandatory to detect wear states inline during blanking or to quantify the actual tool geometry. Therefore, three monitoring approaches can be found in the literature: 1. binary fault detection (BFD), 2. feature-based supervision (FS) and 3. machine learning-based supervision (MLS).

BFD is mainly used in industrial practice and is limited to the identification of binary fault classes. By setting thresholds, envelopes and windows based on experiential knowledge, a distinction is made between process states that are acceptable or unacceptable (OK / NOK) [25,26].

In the case of FS, features are extracted from sensorial acquired data based on engineering knowledge. In their study, Hohmann et al. presented such a FS approach and showed that engineered features extracted from a force signal correlate with occurring abrasive and adhesive wear [21]. Klingenberg and de Boer extracted a feature describing the length of the punch penetration into the sheet until the maximum force is reached, as well as the work done in the punch phase, and correlated them with the rounding of the cutting edge [27]. A similar approach was chosen by Kubik et al., who extracted four features from the force signal and correlated them with the cutting edge radii of the punch [3,28].

In the MLS approach, features are extracted without the engineering or experiential knowledge. This so called transformation extracts features by an algorithm from the time domain, the frequency domain, the time-frequency domain or it is based on a model [29]. In the following step, these features serve as input for the training procedure of a model. Model types range from simple regression models to complex deep neuronal networks. One of the first applications of the MLS is described by Lee et al. [30]. They used an autoregressive time-series model to extract coefficients from blanking force. Based on these coefficients, the tool state (sharp vs. worn) was predicted using a linear discriminant function. Jin and Shi extracted features from the tonnage signal of a stamping process using a model-based transformation approach. These features served as an input for a hierarchical classifier to detect abnormal process states [31]. Hamblin presented a backpropagation neural network for predicting burr height formation of blanked workpieces. Here, the inputs for the network are generated by a finite element analysis of the blanking process [32]. Ge et al. used a support vector machine (SVM) in their study to detect abnormal health conditions [33]. The entire signal of the stamping process serves as input for the SVM. In his work, Zhou et al. identified missing workpieces in one of the die stations of a progressive tool based on features extracted by principal component analysis (PCA) and SVM [34]. Asahi et al. present a Temporal Convolutional Network autoencoder for extracting features from raw time series and a SVN to classify 3 discrete wear states [35]. A similar framework is presented by Niemietz et al., leveraging the architecture of an autoencoder (AE) to track tool wear progression during fine blanking [36]. Another approach is shown by Molitor et al. who determine the wear state of a blanking tool inline by capturing images of the processed parts. The authors use Deep Convolutional Neural Networks to estimate 16 wear classes with classification accuracies of up to 99% [37].

A summary of the state of the art shows that wear significantly affects the efficiency of blanking processes. This applies in particular to abrasive wear, which occurs more and more frequently due to the processing of high strength materials. Though techniques such as FS and MLS are described in the literature for in-line wear state estimation, monitoring approaches in practice are limited to manual inspections and BFD. Although procedures for inline estimation of the actual wear condition based on sensor-acquired data can be found in the literature, a concrete specification (transformation and model) for the MLS approach is not given. In particular, the combination of extracted features and the chosen model and their influence on the estimation quality is missed. In addition, there is no information on the absolute computing time of the different FS and MLS approaches. However, this is important for high speed forming process such as blanking, where the time to process a single product is even less than 100 ms. In order to realize an inline and workpiece related tool state estimation on an edge device, the absolute computing time must to be minimized by maximizing the performance of the model.

4. Methodology

In order to estimate the actual abrasive wear state during blanking via DBM inline, three principal steps must be completed. In the first step, data is acquired to form the process as a basis for a further transformation and modelling procedure. Process data (time series: force) and quality data (label: wear
state), which represent the label for each time series, are acquired. In the second step, the amount of data is reduced by a transformation (feature extraction) and serves as input for the modelling procedure. Since the BFD does not inherently allow a classification of multiple instances, this approach is not considered for further investigations in this study. For the FS approach, 10 engineering features are combined with 10 classification models. For the MLS approach, 23 transformation algorithms are combined with 10 classification models. As a result, 10 trained models for the FS approach and 230 trained models for the MLS approach are available, allowing classification of 16 wear classes based on the blanking force. In the third step, each model from the FS and MLS approaches is evaluated based on the key parameters showing its suitability for a decentralized monitoring of blanking process at the edge. Figure 3 summarizes the procedure.

**Figure 3:** Procedure for quantifying the key parameters for a classification of 16 wear states

4.1. **Step 1: Experimental procedure and data acquisition**

All experiments were carried out on a Bruderer AG (BSTA 810) high-speed press at a stroke height of 35 mm and stroke speed of speeds of 100 spm and 300 spm. The blanking tool selected is a cylindrical punch with a diameter of 6 mm and a clearance of 0.15 mm. A cold rolled steel DC03 (1.0347) with a sheet thickness of $2 \pm 0.02$ mm and a tensile strength of $299.8 \pm 2.9$ MPa is selected as the test material. As shown in Figure 2 (a), the blanking tool consists of a lower and an upper part, which are connected by four guiding columns. Thereby, the blanking force is acquired via a piezoelectric force washer (Kistler 9051A) integrated into the direct force flux. The voltage signal of the sensor is digitized by a CompactRIO (NI cRIO 9047) with integrated measuring modules (NI 9215 – analog voltage input $\pm 10$ V) at a sampling frequency of $f_s = 50$ kHz. The final measuring range during the blanking cycle is limited to an angle from $150^\circ$ to $210^\circ$ of the ram. This leads to approximately $10550 / 3520$ data points per time series at a stroke speed of $100 / 300$ spm. Since the data set for each wear state consists of approximately 123 observations, the force matrix is described by $F \in \mathbb{R}^{m \times n}$ ($F_{100} \in \mathbb{R}^{1968 \times 10550}$ and $F_{300} \in \mathbb{R}^{1968 \times 3520}$), where $m$ is the total number of observations for all wears states and $n$ is the number of data points per time series. The labels for the modeling step are described by the cutting edge radii $r_1$ of the punch which quantifies the actual abrasive wear state. They are varied in steps of $0.05$ mm, resulting in 16 labeled classes $\gamma \in \{0, ..., 0.75\}$ mm. Since a critical wear state at the cutting edge has to be specified individually for each tool configuration and depends on the resulting geometry of the blanked workpiece (e.g. burr height [38]), a range up to $r_{16} = 0.75$ mm shall be estimated by the model. This maximum limit of excessive wear is slightly above the cutting edge radii of up to $0.4$ mm shown in the literature [27]. However, it allows the classification model to be adapted to tool configuration with a higher tolerable cutting edge radii.
4.2. Step 2: Modelling and transformation

Based on the acquired data, features are extracted from the matrix $F$, leading to a feature vector $f \in \mathbb{R}^{m \times k}$, where $m$ is the total number of observations and $k$ the number of extracted features with $k \leq n$. Afterwards, the feature vector serves as an input for the classification model, estimating the 16 abrasive wear states. However, to ensure a comparability between each transformation algorithm combined with a classification model, the number of features must be constant. Since for the FS approach ten engineering features are available, the total number of features for the MLS approach is fixed to 10, leading to a feature vector $f \in \mathbb{R}^{10 \times 10}$. While for the FS approach only engineering features are extracted for the MLS approach, 23 transformation algorithms are used. Figure 4 (a) summarizes the transformation algorithms for each approach and gives an overview of the applied classification models. In the MLS approach, features are extracted in the time domain, time-frequency domain or model-based. In the time domain, 10 engineering features according to Hohmann et al. are directly extracted from each time series [21]. Therefore, a force signal is divided into three phases, characteristic points are identified within each phase and features are derived describing the length ($l_{\text{punch}}$, $l_{\text{push}}$, $l_{\text{withdraw}}$), the maximum force ($F_{\text{max,punch}}$, $F_{\text{max, push}}$) and minimum force ($F_{\text{min, withdraw}}$), the work done ($W_{\text{punch}}$, $W_{\text{push}}$ and $W_{\text{withdraw}}$) as well as the elastic gradient during the punch phase ($\delta_{\text{punch}}$).

From the time-frequency domain, the wavelet transformation (WT), Hilbert Huang transformation (HHT) and Wigner Ville distribution (WVD) are used to extract features, transforming the one-dimensional signals into a two-dimensional function of time and frequency [39,40]. While the HHT and the WVD have a fixed mathematical algorithm, the wavelets in the WT are varied according to the coefficients Haar (WThaar), Daubechies with momentum 1 (WTdeb1), momentum 10 (WTdeb10) and momentum 20 (WTdeb20) and Coiflet with momentum 1 (WTcoi1), momentum 3 (WTcoi3) and momentum 5 (WTcoi5) [40].

(a) combined transformation algorithm and classification model

(b) calculation of absolute computing time

![Image](https://via.placeholder.com/150)

**Figure 4:** 240 combinations of transformation algorithms and classification models spread over FS and MLS approaches (a) and procedure for calculating the absolute computing time depending on the configuration $T_{\text{total}}$ (transformation based on the total data set) and $T_{\text{time}}$ (transformation based on a single time series) (b)

Matrix factorization, manifold learning, random projection, as well as ML based models discriminant analysis and AE are used for the model-based transformation. The matrix factorization finds latent structures in a data base and represents them in a lower dimensional space. Representatives of matrix factorization techniques are the principal component analysis (PCA) and its variants kernel PCA.
(KPCA) and sparse PCA (SPCA) [41]. While ordinary PCA is a linear method describing its principal components by linear combinations of all input variables, SPCA overcomes this disadvantage by finding linear combinations that contain only a few input variables by adding a constraint on the number of non-zero elements in the matrix [42]. In KPCA, the data is projected by a kernel function into a higher dimension where it is linearly separable. In this study linear (KPCAlin), quadratic (KPCApoly), sigmoid (KPCAsig) and cosine (KPCAscos) kernels are used.

Manifold learning techniques transfer data from a multidimensional space into a two dimensional space via a non-linear mapping. The algorithms considered in this paper are multidimensional scaling (MDS), linear locally embedding (LLE), locality preserving projection (LPP), t-distributed stochastic neighbor embedding (TSNE) and the uniform manifold approximation and Projection (UMAP) [43].

In addition to the model-based approaches, a sparse random projection transformation (SRP) [44], a linear discriminant analysis (LDA) [45] and an AE are used for feature extraction in this study.

As there are many different types of ML methods used for various applications, the transformation techniques are combined with 10 classification models. Therefore, in this study a Naive Bayes classifier (NBC), classification and regression trees (CART), random forests (RF), linear (LDA) and quadratic discriminant analysis (QDA), multiple logistic regression (MLR), k-nearest neighbor (kNN) and support vector machines (SVM) are used to estimate the abrasive wear state with a minimal error. Since the SVM depends on a kernel function that transfers the input into a higher dimensional space, a polynomial (SVMpoly), a sigmoid (SVMsig) and a radial basis (SVMrbf) function are used in this study.

4.3. Step 3: Training procedure

For the training process of the 240 classification models, the sensorial acquired data is divided into 80% training data and 20% test data and statistically validated by 50 training cycles. It shows that the time for the training procedure of the classification models is independent of the transformation step which is demonstrated by a row dominance in Figure 5. In particular, models such as LDA and QDA as well as KNN and NBC require less training time. Moreover, they are easy to implement since no hyperparameter optimization has to be conducted. In contrast, SVMs show a training effort that is up to 10 times higher and requires a more or less complex hyperparameter optimization depending on the kernel function used. However, the training process for the investigated classification models is associated with low computational effort and training times of less than 1.5 seconds.

![Figure 5: Summarizing the absolute time required to train the classifications model depending on the transformed data](image)

4.4. Step 4: Evaluating efficiency of the DBM via key parameters

In order to compare the efficiency of the DBM, the key parameters model performance and absolute computing time are considered. Since various classification models determine a distribution of classes (e.g. NBC), the receiver operating characteristics (ROC) is used to calculate the model performance depending on its probability distribution over the classes [46]. To compare the ROCs for different models, the area under the curve (AUC) is calculated as a key parameter for the classification model performance. A higher value for the AUC close to 1 indicates that the classes are highly separable by
the model. Since a multiple classification problem is considered in this study, \( N \) number of AUC are set up for \( N \) classes considering the one versus all methodology. The resulting performance parameter is the mean value of the AUC for each class.

Since the absolute computing time is affected by a number of parameters, the strategy shown in Figure 4 (b) is used to quantify this value. Thus only a part of the total computing time provided by the central processing unit (CPU) is available for the execution of the DBM procedure. Other running programs, such as the task manager, browser or background processes, are excluded from the computational time. This also includes the time for reading and writing to the random access memory (RAM), which must be taken into account due to the memory size of the input excess 16 MB of the CPU’s internal memory. This makes 50 % of the CPU power available for the DBM process consisting of transformation and modelling. The CPU time is calculated by Equation (1), where \( n_{\text{IC}} \) is the number of clock cycles for executing the DBM and \( f_{\text{CPU}} \) is the cycle time of the CPU. Since the absolute CPU time \( t_{\text{CPU,abs}} \) depends on the number of cores used in parallel \( n_c \) and the CPU load \( A \) it can be assumed as follows:

\[
t_{\text{CPU}} = \frac{n_{\text{IC}}}{f_{\text{CPU}}} \quad t_{\text{abs}} = \frac{t_{\text{CPU}}}{n_c A}
\]

In order to transfer the results obtained in this study to other computers, the following empirical approach (2) is chosen. It allows the estimation of the required resources, defined by the absolute computing time \( t_{\text{abs}} \) of the edge device for inline estimation of wear in high speed sheet metal forming processes:

\[
t_{\text{abs}} = t_{\text{abs,study}} \cdot \frac{n_{\text{C,study}}}{n_c} \cdot \frac{A_{\text{study}}}{A} \cdot \frac{f_{\text{CPU,study}}}{f_{\text{CPU}}} = t_{\text{abs,study}} \cdot \frac{3.4}{n_c A f_{\text{CPU}}} \cdot [\text{GHz}]
\]

### Table 1: Computer resources used for the examination in this study

| type        | cores \( n_{\text{C,study}} \) | clock \( f_{\text{CPU,study}} \) | physical RAM | average load \( A_{\text{study}} \) |
|-------------|-------------------------------|----------------------------------|---------------|----------------------------------|
| Intel core i7 4770 | 2                             | 3.4 GHz                          | 16GB          | 50 %                             |

Thus, the calculation of the absolute computing time in this study \( t_{\text{abs,study}} \) for the inline estimation of the wear state is composed of the transformation and insertion of these features into a trained classification model. The training process and the transformation matrix \( R \) generated from the model-based transformation is assumed to be completed. This results in two methods to determine the absolute computing time for the DBM procedure. If the feature extracting time is based on the whole matrix \( F \), this is called a transformation from the total data set \( (T_{\text{total}}) \). If extraction time is based on each time series, this is called a time series based transformation \( (T_{\text{time}}) \). In this case, \( R \) is provided within the training procedure, resulting in a feature vector for the \( n \)-the processed workpiece \( f^{(n)} = R \cdot x^{(n)} \) with \( f \in \mathbb{R}^{(n)10\times1} \), \( R \in \mathbb{R}^{10\times10} \) and \( x \in \mathbb{R}^{(n)10\times1} \) (see Figure 4 (b)).

### 5. Results

The following results quantify the key parameters for real-time estimation of the wear state during blanking considering the absolute computing time, represented by the absolute computing time \( t_{\text{abs,study}} \), and the model performance, represented by the classification accuracy AUC. As explained in section 4.3, the absolute time is summed from the time to transform the entire data set \( (T_{\text{total}}) \) and the time to refeeding it into the model or from the time to transform a single time series \( (T_{\text{time}}) \) and the time to refeeding it into the model. The configuration \( T_{\text{time}} \) does not take into account the computing time for the modelling procedure as well as the generation of the model-based transformation matrix \( R \). Figure 6 (a) shows the absolute computing time plotted against model performance for a sampling frequency of 50 kHz for the configuration for \( T_{\text{total}} \) at 100 spm (10550 data points for each time series). Regardless of the combination of transformation algorithm and classification model, most MLS and FS
approaches reach an AUC value above 97%. In contrast, the absolute computing time depends on the combination of transformation algorithm and classification model, and Figure 6 (b) shows the transformation time is higher by 10^3 on average. The least extraction time is required for a random projection. Here, $F \in \mathbb{R}^{m \times n}$ is multiplied by a randomly generated matrix $R \in \mathbb{R}^{n \times k}$ and the dimension is reduced from $n$ to $k$, so that the computing time is limited to a single matrix multiplication. The computing time for the matrix factorization approach based on an PCA rises since $F$ needs to be decomposed by the sum of eigenvectors of the covariance matrix of $F$. In this case, the eigenvalue problem must be solved, which leads to a higher absolute computing time. In the case of a KPCA, in a previous operation $F$ is transposed into a higher dimensional space to generalize the approach to a non-linear method. This increases the computing time by solving the eigenvalue problem and transferring $F$ to a higher dimensional space.

a) key performance parameters (50 kHz, 100 spm and $T_{\text{total}}$)

| Transformation Algorithm | Absolute Computing Time ($t_{\text{abs,study}}$) |
|--------------------------|-----------------------------------|
| SRP                      | 10^1                              |
| AE                       | 10^2                              |
| WT                       | 10^3                              |
| FeatEng                  | 10^4                              |
| PCA                      | 10^5                              |
| LLE                      | 10^6                              |
| KPCA                     | 10^7                              |
| LDA                      | 10^8                              |
| UMAP                     | 10^9                              |
| TSNE & HHT               | 10^10                             |

b) absolute computing time for transformation algorithm and classification model

| Classification Model      | Absolute Computing Time ($t_{\text{abs,study}}$) |
|---------------------------|-----------------------------------|
| SVM                       | 10^1                              |
| MDS                       | 10^2                              |
| LDA                       | 10^3                              |
| KPCA                      | 10^4                              |
| LLE                       | 10^5                              |
| UMAP                      | 10^6                              |
| TSNE & HHT                | 10^7                              |

Figure 6: Key performance parameters for the $T_{\text{total}}$ configuration (a) and the absolute computing time for different transformation algorithm as well as the time estimating the wear state by different classification models (b)

The transformation time further increases in manifold learning approaches, since they are unsupervised estimators that search for low-dimensional manifolds embedded in high-dimensional space. In doing so, they must iteratively solve a minimization problem to obtain the distances between data points in the high-dimensional matrix $F$ and the matrix mapped to the low-dimensional space. In addition, advanced manifold learning approaches (UMPA and TSNE), attempt to minimize the divergence between two probability distributions provided by input vectors of a high-dimensional
matrix compared to the low-dimensional space [47]. By iteratively solving the minimization problem, manifold learning tends to increase absolute computing time. Particularly advanced approaches exceed the absolute computing time (TSNE: 482.44 seconds and UMAP: 42.71 seconds) of other transformation algorithms.

The ML based AE turns out to be another fast algorithm for feature extraction. The structure of the AE is composed of 2 layers (input layer and compression layer) and intermediate activation functions. Here, $F$ is compressed to 10 perceptrons by the compression layer. Since the learning procedure of the AE is based on a simple mathematical operation, multiplying the input with an activation function, the absolute computing time is low. On the other hand, LDA based on a ML classification algorithm divides $F \in \mathbb{R}^{m \times n}$ into $k$ classes. An attempt is made to find a linear transformation $W \in \mathbb{R}^{m \times k}$ that maximizes the distance between the means of all classes and while minimizing the variance within a class, resulting in a higher absolute computing time.

![Figure 7: Key performance parameters for the $T_{time}$ configuration](image)

Feature extraction from the time and time-frequency domain requires less absolute computing time than model-based transformations such as matrix factorization and manifold learning methods. This is due to the fact that the absolute computing time for matrix operations increases exponentially with the number of data points, while directly extracting features from a single time series increases approximately linearly with the number of data points. Although the feature extraction in the time-frequency domain requires a pre-transformation into the frequency domain, the absolute computing time is similar to the direct extraction of engineering features. Since the engineering features are based on characteristic points determined in a time series similar to the time-frequency transformation, the FS approach is two-stage.

Implementing the DBM in an industrial application requires less absolute computing time, since the transformation procedure is limited to a single time series ($T_{time}$). Figure 7 shows the absolute computing time plotted against the model performance for a sampling frequency of 50 kHz for the configuration for $T_{time}$ at 100 spm (10550 data points for each time series). Compared to the configuration $T_{total}$, the time for processing a stroke related time series is reduced by a factor of $10^3$ while the model performance is still close to 100%. In this context, the model-based transformation is suitable for a real-time application at the edge since the AUC is above 99.99% and the absolute computing time is lower than 80 ms. This is because the time for the model-based transformation is reduced to a matrix multiplication, where a transformation matrix $R$, determined during the training procedure of the model, is multiplied with the matrix $F$. For example, in the case of PCA-based feature extraction, approximately 4.7 ms are available for a workpiece related estimation of the wear state. Assuming a stroke angle of $150^\circ$ to $210^\circ$, Figure 8 (a) shows the total time available for workpiece related processing of a single time series. As the stroke speed increases, the available data processing time decreases exponentially. Thus, by means of PCA features, a workpiece related estimation of the wear state is possible up to stroke speeds of 880 spm. This does not take into account that at a constant
sampling frequency, the number of data points is further reduced for each time series. Not only can the sampling frequency be kept constant, but it can also be reduced further to reduce the absolute computing time. However, it must be verified that the characteristics of the sensorial acquired data still represent the physical state of the process and that the model performance remains at a high level. To this end, this study investigates the dependence of model performance and absolute computing time related to the sampling frequencies (number of samples).

![Figure 8: Available time for a workpiece related data processing depending on the stroke speed (a) and the absolute computing time (b) as well as the AUC value (c), depending on the sampling frequency (number of samples) at 300 spm comparing FS and MLS approaches.](image)

Figure 8 (b) and (c) show the key parameters for a combination of engineering features with a QDA (FS approach) and model-based PCA features combined with a QDA (MLS approach) at a stroke speed of 300 spm for the $T_{\text{time}}$ configuration. Since the processing time for the model-based features stay constant over the number of samples, the absolute computing time for the extraction of the engineering features decreases. At 1680 samples, which corresponds to a sampling frequency of approximately 25 kHz, the engineering features are at an advantage compared to the model-based approach. Looking at the model’s performance, shows that the sampling frequency can be reduced to 675 samples, which corresponds to a sampling frequency of approximately 9.6 kHz.

6. Summary and outlook
The paper shows that DBM is feasible for real-time estimation of wear states during blanking up to strokes speeds above 880 spm. By reducing the sampling frequency to below 10 kHz, the absolute computing time can be further reduced. Even at these reduced sampling frequencies, most of the investigated FS and MLS approaches achieve model performances above 99.99%. In particular, model-based transformation techniques such as PCA, SRP or AE allow a high estimation quality, while absolute computing time is reduced to a minimum. Furthermore, the authors show that at low sampling frequencies (< 25 kHz), engineering features become advantageous with regard to the computational effort compared to the model-based techniques. To prove the applicability of an edge device for an industrial use case depending on the available computer resources, the study provides an empirical formulation (Equation 2) to estimate the absolute computing time needed for processing data in high speed forming processes.

7. Acknowledgement
The results of this paper are achieved within the project Mittelstand-Digital Zentrum Darmstadt funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK). The authors wish to thank for funding and supporting this project. Furthermore, the authors would like to thank Bruderer AG for providing the high-speed press BSTA 810-145 and Dayton Progress GmbH for providing theblanking tools under challenging circumstances.
References

[1] Pennenkamp J, Glebek R, Henze M, Meisen T, Quix C, Hai R, Gleim L, Niemietz P, Rudack M, Knape S, Eppe A, Trauth D, Vroomen U, Bergs T, Brecher C, Buhrig-Polaczek A, Jarke M and Wehrle K 2019 Towards an Infrastructure Enabling the Internet of Production Int. Conf. on Industrial Cyber Physical Systems (Taipei) pp 31–37

[2] Lange K 1985 Handbook of metal forming (New York: McGraw-Hill)

[3] Kubik C, Hohmann J and Groche P 2021 Exploitation of force displacement curves in blanking-feature engineering beyond defect detection Int. J. Adv. Manuf. Syst. 113 pp 261–278

[4] Dilda V et al 2017 Manufacturing: Analytics unleashes productivity and profitability (McKinsey & Company)

[5] Wuest T, Weimer D, Irgens C and Thoben K D 2016 Machine learning in manufacturing: advantages, challenges, and applications Prod. Manuf. Res. 4 pp 23–45

[6] Hoppe F, Hohmann J, Knoll M, Kubik C and Groche P 2019 Feature-based Supervision of Shear Cutting Processes on the Basis of Force Measurements: Evaluation of Feature Engineering and Feature Extraction Procedia Manuf. 34 pp 847–856

[7] Kubik C, Knauer S M and Groche P 2021 Smart sheet metal forming: importance of data acquisition, preprocessing and transformation on the performance of a multiclass support vector machine for predicting wear states during blanking J. Intell. Manuf. 33 p 259–282

[8] Hirsch M, Demmel P, Golle R and Hoffmann H 2011 Light Metal in High-Speed Stamping Tools KEM 473 pp 259–266

[9] Groche P, Hoppe F, Hesse D and Calmano S 2016 Blanking-bending process chain with disturbance feed-forward and closed-loop control J. Manuf. Proc. 24 pp 62–70

[10] Shi W, Cao J, Zhang Q, Li Y and Xu L 2016 Edge Computing: Vision and Challenges IEEE Internet Things J. 3 pp 637–646

[11] Pan J and McElhannon J 2018 Future Edge Cloud and Edge Computing for Internet of Things Applications IEEE Internet Things J. 5 pp 439–449.

[12] Varghese B, Wang N, Barbhuviya S, Kilpatrick P and Nikolopoulos D S 2016 Challenges and Opportunities in Edge Computing Int. Conf. on Smart Cloud (New York) pp 20–26

[13] Al-Turjman F 2019 Edge Computing: From Hype to Reality (Springer International Publishing)

[14] Kitamura K, Makino T, Nawa M and Miyata S 2016 Tribological effects of punch with micro-dimples in blanking under high hydrostatic pressure CIRP Annals 65 pp 249–252

[15] Hambli R 2001 Blanking tool wear modeling using the finite element method Int. J. Mach. Tools Manuf. 41 pp 1815–29

[16] Archard J F and Hirst W 1956 The wear of metals under un lubricated conditions Proc. of the R. Soc. of Lond. 236 pp 397–410

[17] Mucha J 2010 An experimental analysis of effects of various material tool’s wear on burr during generator sheets blanking Int. J. Adv. Manuf. Technol. 50 pp 495–507

[18] Mangalore Babu S S, Berry S, Ward M and Krzyzanowski M 2018 Numerical investigation of key stamping process parameters influencing tool life and wear Proc. Manuf. 15 pp 427–435

[19] Hambli R 2002 Design of Experiment Based Analysis for Sheet Metal Blanking Processes Optimisation Int. J. Adv. Manuf. Syst. 19 pp 403–410.

[20] Cheung C F, Lee W B and Chiu W M 2000 An investigation of tool wear in the dam-bar cutting of integrated circuit packages Wear 237 pp 274–282

[21] Hohmann J, Schatz T and Groche P 2017 Intelligent Wear Identification Based on Sensory Inline Information for a Stamping Process Proc. of 5th Int. Conf. on Advanced Manufacturing Engineering and Technologies (Belgrade) pp 285–295

[22] Cheon S and Kim H 2016 Prediction of tool wear in the blanking process using updated geometry Wear 352-353 pp 160–170

[23] Falconnet E, Chambert J, Makich H and Monteil G 2015 Prediction of abrasive punch wear in copper alloy thin sheet blanking Wear 338-339 pp 144–154
[24] Hoffmann H, Nürnberg G, Ersoy-Nürnberg K and Herrmann G 2007 A new approach to determine the wear coefficient for wear prediction of sheet metal forming tools Prod. Eng. 1 pp 357–363
[25] Marposs Monitoring Solutions GmbH Brankamp 2020 X3 - All-in-one Überwachung für die Metallumformung (http://www.brankamp.com/de/data/files/produkte/prospekt_x3_de.pdf)
[26] Anasagasti M, Godoy L, Socorro R, Garcia R, Iturrospe A and Viguera M 2016 MANTIS project: D1.2 Consolidated State-of-the-Art of Sensor (ECSEL Joint Undertaking)
[27] Klingenberger W and de Boer T W 2008 Condition-based maintenance in punching/blanking of sheet metal Int. J. Mach. Tools Manuf. 48 pp 589–598
[28] Liewald M, Bergs T, Groche P, Behrens B-A, Briesenick D, Müller M, Niemietz P, Kubik C and Müller F 2022 Perspectives on data-driven models and its potentials in metal forming and blanking technologies Prod. Eng. Res. Devel. 7
[29] Li C J Signal Processing in Manufacturing Monitoring 2006 Condition Monitoring and Control for Intelligent Manufacturing vol 21, ed L Wang and R X Gao (London: Springer) pp 245–265
[30] Lee W B, Cheung C F, Chiu W M and Chan L K 1997 Automatic supervision of blanking tool wear using pattern recognition analysis Int. J. of Mach. Tools Manuf. 37 pp 1079–95
[31] Jin J and Shi J 2000 Diagnostic Feature Extraction From Stamping Tonnage Signals Based on Design of Experiments J. Manuf. Sci. Eng. 122 pp 360–369
[32] Hambli R 2002 Prediction of burr height formation in blanking processes using neural network Int. J. Mech. Sci. 44 pp 2089–2102
[33] Ge M, Du R, Zhang G and Xu Y 2002 Fault diagnosis using support vector machine with an application in sheet metal stamping operations MSSP 18 pp 143–159
[34] Zhou S and Jin J 2005 Automatic feature selection for unsupervised clustering of cycle-based signals in manufacturing processes IIE Transactions 37 pp 569–584
[35] Asahi S, Karadogan C, Tamura S, Hayamizu S and Liewald M 2021 Process data based estimation of tool wear on punching machines using TCN-Autoencoder from raw time-series information IOP Conf. Ser. Mater. Sci. Eng. 1157 p 12078
[36] Niemietz P, Unterberg M, Trauth D and Bergs T 2021 Autoencoder based Wear Assessment in Sheet Metal Forming IOP Conf. Ser. Mater. Sci. Eng. 1157 p 12082
[37] Molitor D A, Kubik C, Hetfleisch R H and Groche P 2022 Workpiece image-based tool wear classification in blanking processes using deep convolutional neural networks Prod. Eng. Res. Devel.
[38] Feistle M, Koslow I, Krinninger M, Golle R and Volk W 2017 Reduction of Burr Formation for Conventional Shear Cutting of Boron-alloyed Sheets through Focused Heat Treatment Proc. CIRP 63 pp 493–498
[39] Caesarendra W and Tjahjowidodo T 2017 A Review of Feature Extraction Methods in Vibration-Based Condition Monitoring and Its Application for Degradation Trend Estimation of Low-Speed Slew Bearing Machines 5 p 21
[40] Hosameldin A and Asoke K. N 2019 Condition Monitoring with Vibration Signals: Compressive Sampling and Learning Algorithms for Rotating Machines, (New Jersey: Wiley-IEEE Press)
[41] Bart H et al 2008 Factorization of Matrix and Operator Functions: The State Space Method, (Basel: Birkhäuser)
[42] Bonaccorso G 2018 Machine Learning Algorithms Second Edition, (Birmingham: Packt Publishing)
[43] Ma Y and Fu Y Manifold Learning Theory and Applications, (Boca Raton: CRC Press)
[44] Li P, Hastie T J and Church K W 2006 Very sparse random projections in Proc. of the 12th Int. Conf. on Knowledge discovery and data mining (New York) pp 287-296
[45] Gu Q, Li Z and Han J Linear Discriminant Dimensionality Reduction in Machine Learning and Knowledge Discovery in Databases, vol. 6911 pp 549–564, Springer, Heidelberg 2011.
[46] Huang J and Ling C X 2005 Using AUC and accuracy in evaluating learning algorithms IEEE Trans. Knowl. Data Eng. 17 pp 299–310
[47] van der Maaten L 2014 Accelerating t-SNE using tree-based algorithms *J. Mach. Learning Research* pp 3221–45