The International Journal of Advanced Manufacturing Technology (2019) 104:3647–3662
https://doi.org/10.1007/s00170-019-04090-6

ORIGINAL ARTICLE

Tool wear classification using time series imaging and deep learning

Giovanna Martínez-Arellano[1] · German Terrazas[2] · Svetan Ratchev[1]

Received: 18 October 2018 / Accepted: 30 June 2019 / Published online: 17 July 2019

Abstract
Tool condition monitoring (TCM) has become essential to achieve high-quality machining as well as cost-effective production. Identification of the cutting tool state during machining before it reaches its failure stage is critical. This paper presents a novel big data approach for tool wear classification based on signal imaging and deep learning. By combining these two techniques, the approach is able to work with the raw data directly, avoiding the use of statistical pre-processing or filter methods. This aspect is fundamental when dealing with large amounts of data that hold complex evolving features. The imaging process serves as an encoding procedure of the sensor data, meaning that the original time series can be re-created from the image without loss of information. By using an off-the-shelf deep learning implementation, the manual selection of features is avoided, thus making this novel approach more general and suitable when dealing with large datasets. The experimental results have revealed that deep learning is able to identify intrinsic features of sensory raw data, achieving in some cases a classification accuracy above 90%.

Keywords Smart manufacturing · Tool wear classification · Time series imaging · Convolutional neural network · Deep learning

1 Introduction

The manufacturing industry has gone through several paradigm changes along the years. Industrie 4.0, also referred as smart industry, is a new paradigm that proposes the integration of information and communication technologies (ICT) into a decentralised production. With manufacturing machines fully networked to share data and controlled by advanced computational intelligence techniques, this paradigm is looking to improve productivity, quality, sustainability and reduce costs [1, 2].

✉ Giovanna Martínez-Arellano
giovanna.martinezarellano@nottingham.ac.uk

German Terrazas
gt401@cam.ac.uk

Svetan Ratchev
svetan.ratchev@nottingham.ac.uk

1 Institute for Advanced Manufacturing, University of Nottingham, Nottingham, UK

2 Institute for Manufacturing, University of Cambridge, Cambridge, UK

The estimation of the remaining useful life (RUL) of industrial components is an important task in smart manufacturing. Early detection of cutting tool degradation facilitates the reduction of failures, and hence decreases manufacturing costs and improves productivity. It can also help maintain the quality of the workpiece, as it has been demonstrated that there is a correlation between the surface roughness of the workpiece and the cutting tool wear [3]. Real-time tool wear measurement is difficult to put in practice as the tool is continuously in contact with the workpiece during machining. For this reason, a plethora of indirect approaches for tool wear estimation (also referred as *Prognosis*) have been proposed utilising sensor signals such as cutting forces, vibrations, acoustic emissions and power consumption [4].

Prognostic approaches can be divided into two categories: model-based and data-driven. The first ones rely on the a priori knowledge of the underlying physical laws and probability distributions that describe the dynamic behaviour of a system [5–8]. Although these have proven to be successful, an in-depth understanding and expertise of the physical processes that lead to tool failure is required.

On the other hand, data-driven approaches model the data by means of a learning process, avoiding any assumptions

on its underlying distribution. Most data-driven methods that have been used for tool wear prediction are based on machine learning, particularly artificial neural networks (ANN), support vector machines (SVM) and decision trees (DT) [9]. However, these techniques are limited in their ability to process raw (i.e. unstructured or unformatted) data, which has a negative effect on their generalisation capabilities [10].

The large amount of data in smart manufacturing imposes challenges such as the proliferation of multivariate data, high dimensionality of feature space and multicollinearity among data measurements [2, 11]. This paper presents in detail the methodology of a novel approach for tool wear classification recently used in [12] as a component of an on-line monitoring framework. Its automatic feature learning and high-volume processing capabilities make deep learning a viable advanced analytics method for tool wear classification despite the large volumes of data required. The proposed classification methodology is based on two components: an imaging step and a deep learning step. The imaging technique employed encodes sensor signals in such a way that its complex features as well as the exhibited temporal correlations are captured by the deep learning, avoiding manual selection. An analysis of the challenges and strategies used to build a big data classifying approach is performed through a set of experiments using the PHM 2010 challenge dataset [13], where the technical procedures of how the data was generated and collected are not entirely known. This provides a way to perform an unbiased blind test and proof of the generalisation capabilities of the methodology.

The rest of the manuscript is organised as follows: Section 2 presents details of how machine learning has been applied to tool wear prediction. Section 3 introduces the proposed approach giving details of the signals imaging and the deep learning methodology. The experimental setup and the results and discussion are presented in Section 4. Finally, conclusions and future work are presented in Section 5.

2 Related work

Tool wear has been widely studied as it is a very common phenomenon in manufacturing processes such as milling, drilling and turning. It is well known that different machining parameters such as spindle speed, feed rate and cutting tool characteristics as well as the workpiece material have an effect on tool wear progression [14]. Although this progression can be mathematically estimated [15, 16], these models rarely capture the stochastic properties of real machining processes and tool-to-tool performance variation [17]. Over the last two decades, it has been demonstrated that data-driven models can achieve higher accuracy, although these have also shown some drawbacks [10].

Some of the most common data-driven methods are based on traditional machine learning algorithms. SVMs, for example, have been successfully applied for tool condition monitoring in [18]. The authors use automatic relevance determination (ARD) on acoustic emission data to select nine features as inputs for classification. ANNs have also been extensively applied for tool wear prediction. These commonly use a combination of cutting parameters such as cutting speed, feed rate and axial cutting length as well as statistical features of forces, vibrations and acoustic emission [19–22]. In applications such as drilling and milling, it has been shown how ANNs can outperform regression models. In [9], a tool wear prediction method based on random forests is proposed. Although this approach has outperformed ANN- and SVM-based methods, it relies on the manual selection of features to build the internal classification structures.

Manual feature selection is a significant problem when dealing with large amounts of shop floor–generated sensory data. Its distribution as well as the number of features available may change with time. Cloud-based architectures recently proposed for collecting and managing sensory data [2, 23] present new challenges to current TCM solutions. To develop a more general approach, forthcoming approaches should be able to cope not only with high volumes of heterogeneous data but also with the constant evolution of high-dimensional features. Most classical machine learning techniques have been designed to work with data features that do not change with time (static data). As a result, several of these techniques either have been extended to handle the temporal changes or rely on a prior selection of features using other algorithms [24].

Deep learning has offered better solutions when dealing with high-dimensional evolving features. These techniques have made major advances in fields such as image recognition [25, 26], speech recognition [27] and natural language processing [28, 29], to name a few. Its capability to process highly complex featured data has led to an emerging study of deep learning applications for smart manufacturing. For instance, recurrent neural networks (RNN) have been successful for the long-term prognosis of rolling bearing health status [30]. In [31], a local feature-based gated recurrent unit network is applied to tool wear prediction, gearbox fault diagnosis and bearing fault detection. The bi-directional recurrent structure proposed by the authors can access the sequential data in two directions–forward and backward–so that the model can fully explore the ‘past and future’ of each state.

Another successful deep learning architecture is the convolutional neural network (CNN) [32], which is the

one addressed in this work. CNNs have become the de facto standard for deep learning tasks as they have achieved state-of-the-art performance in image recognition tasks. The architecture of a CNN is based on the architecture of the ANN, but further extended with a combination of convolutional and sub-sampling layers that allow the discovery of relevant features. This is explained in more detail in Section 3.2. CNNs are developed primarily for 2D signals such as images and video frames. Some successful applications are the detection of vehicles in complex satellite images [33], the classification of galaxy morphology [34], brain tumour segmentation from MRI images [35], among others. Their success in the classification of two-dimensional data has led to further development of CNNs for time series classification (one-dimensional data). Some applications include the classification of electrocardiogram beats for detecting heart failure [36] and the use of accelerometer readings for human activity recognition [37].

CNNs have also been applied in manufacturing problems. For example, this technique has been used for the detection of faulty bearings [38–40] by feeding raw vibration data directly to the CNN, achieving good accuracy and reducing the computational complexity of the extraction of fixed features. In [41], real-time structural health monitoring is performed using 1D CNNs. The authors use vibration signals from damaged and undamaged joints of a girder to train several CNNs, one for each joint. Their objective is to detect the structural damage (if any), and identify the location of the damaged joint(s) in the girder. The authors report an outstanding performance and computational efficiency of the approach when dealing with large-scale experiments.

Some previous work on tool wear prediction using a CNN combined with bi-directional long short-term memory (LSTM) has been done [42]. The proposed approach is able to extract local features of the data, achieving good accuracy when compared with other deep learning techniques such as RNNs. However, the method performs a substantial size reduction of the original data, losing information at the flute level. This will be further discussed in Section 5.

Manual feature selection is still a limitation for tool wear prediction approaches to achieve generalisation. To address this, this paper extends preliminary experiments of a novel deep learning–based method that will allow the automatic discovery of intricate structures in sensor signals that relate to the tool condition, and from this provide a classification of the tool state. The approach is blind to the type of signals given or their underlying distribution, so no assumptions nor manual feature selections are needed. At the same time, the model is blind to the type of wear being classified. Although in this work flank wear has been used as a measure of the tool condition, the proposed methodology could be used for other types of tool wear as well.

3 Methodology

This section presents the two main steps of the methodology: the imaging of sensor signals using Gramian Angular Summation Fields [43] and the classification using CNNs. The idea behind this approach is to visually recognise, classify and learn structures and patterns intrinsic to sensory data without loss of information.

3.1 Time series imaging

There has been a recent interest on reformulating features of time series to improve their identification, and hence classification. Eckmann et al. introduced the method of *recurrence plots* to visualise the repetitive patterns of dynamical systems [44]. Silva et al. used this method and proposed the use of a compression distance approach to compare recurrence plots of time series as a way to measure similarity [45]. Methods based on time series to network mapping using the topology of the network as a way to characterise the time series have also been proposed [46, 47]. Most of these methods do not provide a way to reconstruct the original data, making unclear how the topological properties relate to the time series. Wang et al. propose three techniques, two based on Gramian Angular Fields (GAF) and one on Markov Transition Fields (MTF) to image time series [43]. They argue that compared with previous techniques, the original time series can be re-constructed, allowing the user to understand how the features introduced in the encoding process improve classification. They reported GAF encoding methods were able to achieve competitive results in a series of baseline problems that include different domains such as medicine, entomology, engineering and astronomy. Furthermore, this method has been found to perform well compared with other time series encoding techniques in applications such as the classification of future trends of financial data [48].

As a pre-processing step, our approach uses the GAF imaging technique proposed by [43], particularly the one based on the summation of angular fields, Gramian Angular Summation Fields (GASF). This encoding method consists of two steps. First, the time series is represented in a polar coordinate system instead of the typical Cartesian coordinates. Thus, given a time series $X = x_1, x_2, ..., x_n$ of n real-valued observations, X is rescaled so that all values fall in the interval $[-1, 1]$ by:

$$\tilde{x}^i_{-1} = \frac{x_i - max(X) + (x_i - min(X))}{max(X) - min(X)} \quad (1)$$

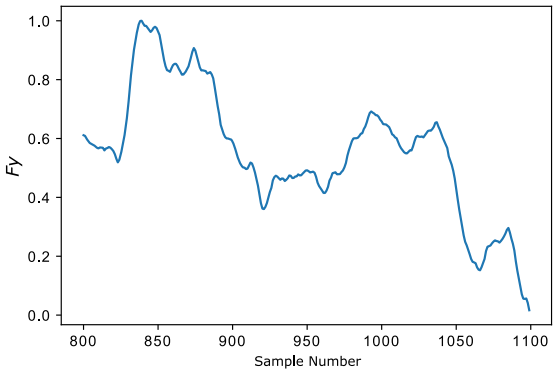
The time series $\tilde{X}$ can then be represented in polar coordinates by encoding the value as the angular cosine and the time stamp as the radius applying Eqs. 2 and 3:

$$\phi = arccos(\tilde{x}_i),\ -1 \leq \tilde{x}_i \leq 1,\ \tilde{x}_i \in \tilde{X} \tag{2}$$

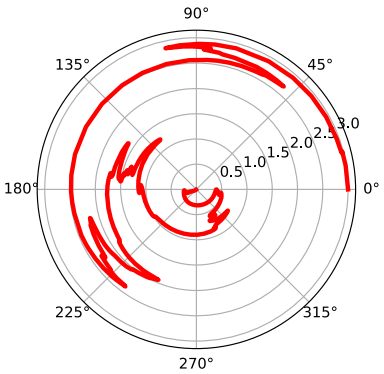
$$r = \frac{t_i}{N},\ t_i \in \mathbb{N} \tag{3}$$

In Eq. 3, t_i is the time stamp and N is a constant factor to regularise the span of the polar coordinate system. Figure 1 shows an example of forces on z-dimension and its representation in polar coordinates.

As time increases, corresponding values on the polar coordinate system warp among different angular points on the spanning circles. This representation preserves the temporal relations and can easily be exploited to identify



(a) Example of a rescaled raw signal from the dynamometer in the y-axis using Equation 1.



(b) Polar coordinates representation of the raw signal shown in (a).

Fig. 1 Forces on y-axis acquired from a dynamometer are encoded as polar coordinates by applying Eqs. 2 and 3. As time increases, the corresponding values of the signal in polar coordinates wrap among different angular points on the spanning circles, keeping the temporal relations

the temporal correlation within different time intervals. This temporal correlation is represented as:

$$G = \begin{bmatrix} cos(\phi_1 + \phi_1) & \dots & cos(\phi_1 + \phi_n) \\ cos(\phi_2 + \phi_1) & \dots & cos(\phi_2 + \phi_n) \\ \vdots & \ddots & \vdots \\ cos(\phi_n + \phi_1) & \dots & cos(\phi_n + \phi_n) \end{bmatrix} \tag{4}$$

$$cos(\phi_i + \phi_j) = \tilde{X}' \cdot \tilde{X} - \sqrt{I - \tilde{X}^2}' \cdot \sqrt{I - \tilde{X}^2} \tag{5}$$

where I is a unit full row vector ([1,1,...,1]). Figure 2 shows the resulting image of applying the encoding method to the time series presented in Fig. 1.

The GASF image provides a way to preserve temporal dependency. Time increases as the position in the image moves from top–left to bottom–right. $G_{(i,j||i-j|=k)}$ represents the relative correlation by superposition of directions with respect to time interval k. The main diagonal $G_{i,i}$ is the special case when $k = 0$, which contains the original value/angular information. The dimension of the resulting GASF image is $n \times n$ when the time series is of length n. To reduce the size of the image, piecewise aggregation approximation (PAA) is applied to smooth the time series while keeping trends [49]. As explained in the Experiments section, the amount of time series data that is acquired from the sensors is large (more than 200,000 measurements), so PAA is fundamental to keep the images at a reasonable size without losing time coherence.

To label the images, three regions have been identified as defined in [50]. According to the literature, the tool life in milling operations is typically divided into three stages/classes: a break-in region, which occurs with a rapid wear rate; the steady-state wear region with uniform wear rate; and a failure region, which again occurs with a rapid wear rate [51]. Figure 3 presents a tool degradation curve example with the classes that were used to label the images.

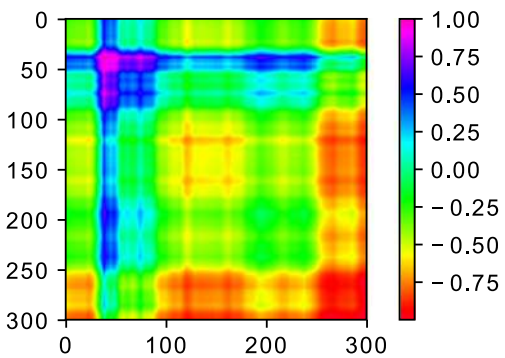


Fig. 2 Example of the encoding of forces in the y-axis as an image using GASF. The colour represents the intensity of the relative correlation between two points in the time series, which is a value between −1 and 1. There is no PAA smothing applied to the resulting image, so the resolution (300 × 300 pixels) is the same as in the original signal

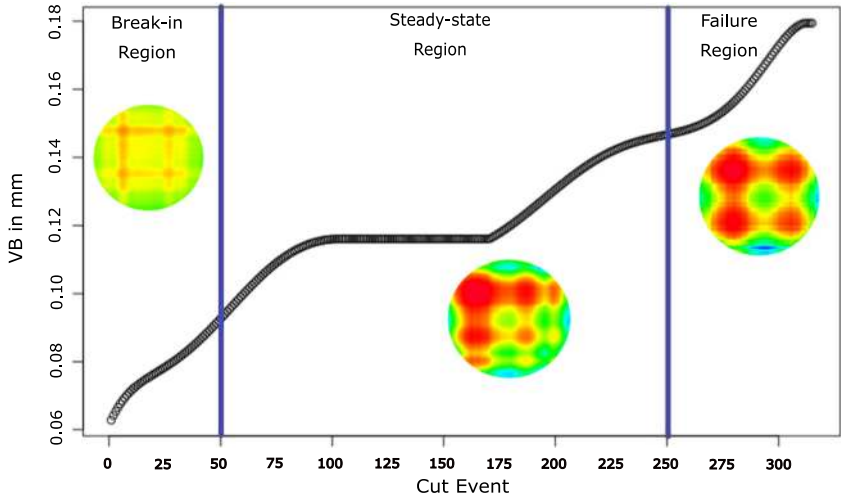


Fig. 3 Tool flank wear as a function of cutting time (cut events of cutter c_6 used in the experiments). For each region, a sample image of forces in y-axis is provided

3.2 Deep learning for time series classification

To identify the current state of wear of a tool by using sensor signals, the approach applied needs to be capable of picking up the temporal dependencies present in the signals. Sensor signals are expected to show changes in their temporal structures as the tool wears out. A classification tool should be capable of identifying those changes and map them to a predefined wear class.

Time series classification methods are generally divided into two categories: sequence-based methods and feature-based methods. Among both of these categories, *k*-nearest neighbour (*k*-NN), which is a sequence-based method, has proven to be very difficult to beat. This is specially true when paired with dynamic time warping (DTW). The drawback of this approach is its lengthy computation time. As the training set grows, the computation time, and hence the prediction time, increases linearly.

An approach that can provide constant prediction time as well as a way to extract relevant features automatically is deep learning. CNNs in particular have been successful in handling large volumes of data. Although they have been primarily used for visual tasks, voice recognition and language processing, new developments have looked towards time series classification.

CNNs have been inspired by the way the visual cortex in the human brain works. Neurons of the visual cortex have a *local receptive field* which reacts to visual stimuli located in a limited region of the visual field [52]. These receptive fields may overlap, tiling together the whole visual field. Some neurons have larger receptive fields which react to more complex patterns that are further combinations of lower level patterns. The discovery of these basic functionalities of the human brain inspired the idea of developing an artificial neural network architecture whereby higher level neurons are based on the outputs of neighbouring lower level neurons, to detect complex patterns. In 1998, LeCun et al. [32] proposed the LeNet-5 architecture, which contains the main building blocks of a CNN: the *convolution layer* and the *pooling layer*.

A convolution layer is formed by a series of neurons that are connected to neurons of a previous layer based on the their receptive field. For example, in the first convolution layer, each neuron is not connected to each individual pixel of the input image, but to only those pixels within a receptive field. Then each neuron in the second convolution layer is connected to neurons within a small rectangle in the first layer. The first convolution layer is responsible for detecting the lower level features, and further convolutions assemble these features into higher level ones. The set of weights (i.e. filter) of a neuron in each convolution layer will depend on the type of feature it is "looking" for. For example, a particular filter would be able to detect vertical lines while another one could detect horizontal ones. During the convolution, the filter is compared with different areas of the image, obtaining a *feature map*, that highlights the areas in an image that are most similar to the filter (see Fig. 4a). As images posses a variety of different features, each convolution neuron would have more than one set of weights or filters. The training process will enable the CNN to find the most useful filters for the particular classification task. In the case of the force classification that is addressed here, the training process will find those filters that allow it to recognise in a first instance features at a flute level regardless of where in the image they are located. Then, higher level convolutions allow the determination of the state of the tool considering all flutes.

The pooling layer is another important building block of the CNN. This layer downscales the output of the convolution, thus reducing dimensionality, the local sensitivity of the network and computational complexity (see Fig. 4b) [32]. A typical CNN architecture stacks

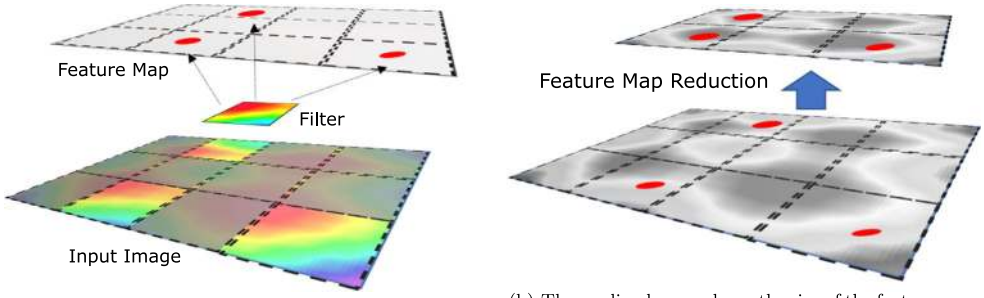


Fig. 4 Low-level features of forces are picked up by the first layer, which are then assembled into higher level features in the following layers

(a) In the convolution step, a feature map is obtained for each filter containing information of the areas where the feature was detected.

(b) The pooling layer reduces the size of the feature maps while keeping the information of the zones where the features were found.

several convolutions (that may include a rectified linear unit (ReLU) step to speed up the training) and pooling layers which reduce the size of the image as it gets deeper. Finally, at the top of the stack, a multilayer neural network is connected to the last convolution/pooling to perform the classification.

In this paper, the CIFAR-10 architecture from Tensorflow has been used [53]. This is an off-the-shelf CNN architecture that has proven to achieve high accuracy on the classification of 3-channel images (see Fig. 5). This architecture has two convolution layers stacked with their corresponding ReLU and pooling layers. Each convolution applies 64 filters. As will be presented in the next section, the implemented CNN will take 3-channel images generated from the force sensors and use these for training. The deep learning structure will be able to pick up the relevant features that relate to tool wear condition. Figure 6 shows a schematic of how the approach has been implemented.

4 Experiments and results

Tool wear classification was performed using a dataset that was originally made available by the PHM2010 Data Challenge [13]. The dataset contains sensory data of six 3-flute cutters (labelled c_1, ..., c_6) used in a high-speed CNC machine (Röders Tech RFM760) under dry milling conditions until a significant wear stage. The experiment with each cutter was carried out as follows. The workpiece surface was machined line-by-line along the x-axis with a 6-mm three-flute cutter. After finishing one pass along the x-axis (axial depth of 0.2 mm and radial depth of 0.125 mm), the tool was retracted to start a new pass. This was done until the complete surface was removed. Then, the tool was removed from the tool holder and taken to a LEICA MZ12 microscope, where the corresponding flank wear (V_b) for each individual flute was measured. In order to capture cutting forces throughout the experiment, a Kistler quartz 3-component platform dynamometer was

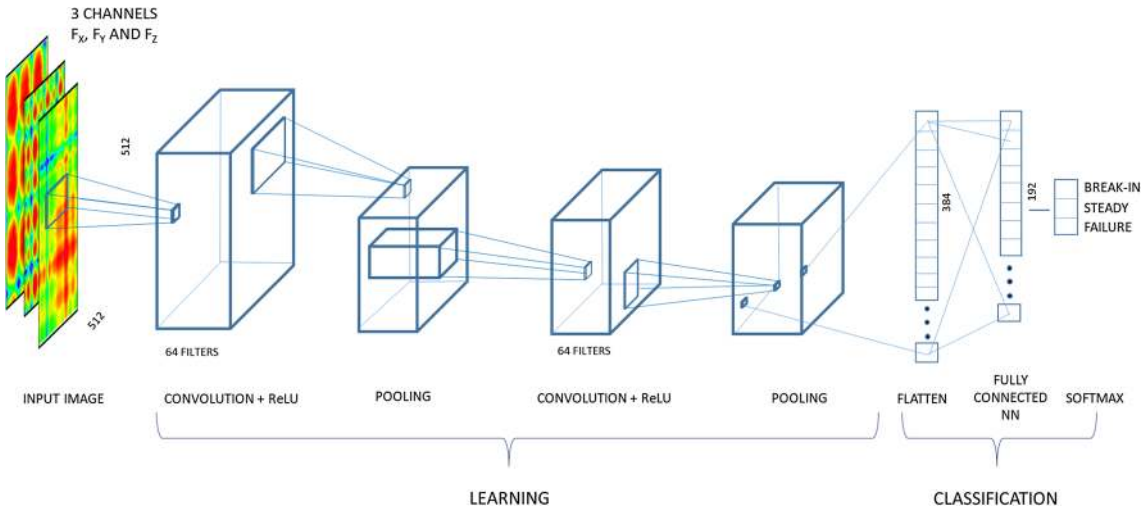


Fig. 5 CNN architecture based on the Tensorflow implementation for the CIFAR-10 dataset (adapted from [53])

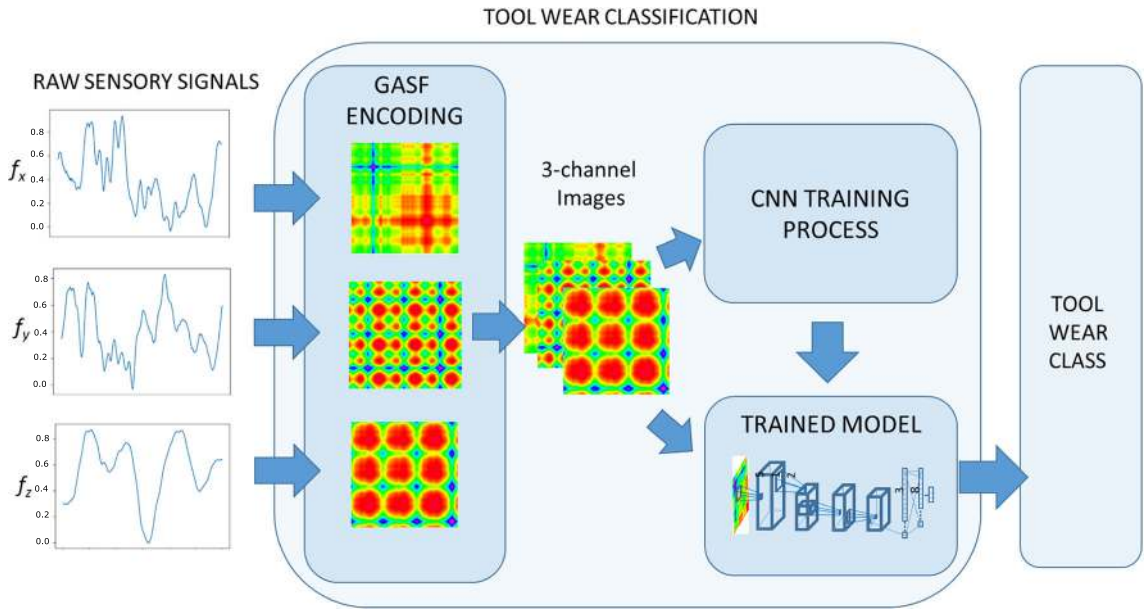


Fig. 6 Framework proposed combining time series imaging and deep learning for tool wear classification. Forces in the three dimensions are individually encoded using GASF and put together as 3-channel images. From those images, 70% is used for training a CNN model and then 30% used for testing

mounted between the workpiece and the machining table. A schematic of this setup is shown in Fig. 7. To measure the vibrations, three Kistler piezo accelerometers were mounted on the workpiece. Finally, an acoustic emission sensor was mounted on the workpiece to monitor the high-frequency stress wave generated by the cutting process. For each cutter, the seven signal channels (forces in the x-, y- and z-axes, vibrations in the x-, y- and z-axes and acoustic emission) were recorded while removing 315 layers of the stainless steel workpiece (see Table 1). Table 2 shows the details of the process conditions during the cutting tests. The total size of the dataset for each cutter is about 3.2 GB, making in total nearly 20 GB for all cutters. In this work, only three of the six cutters (c_1, c_4 and c_6) were used as these were labelled with their corresponding tool wear measurements. More details on the machining setup can be found in [54].

Initial experiments were carried out with a data subset comprising a single cutting tool for the training and test sets, with a total data set size of 1 GB. In this case, the cutter labelled c_6, from which 315 cuts and tool wear measurements are available, was used. Force signals were selected as the only input for the CNN to avoid a computationally expensive training process for this proof of concept.

To prepare the dataset for training and testing of the CNN, each cutting force F_x, F_y and F_z corresponding to a removed layer was encoded as three separate images. Since the time series that corresponds to one layer can be as long as 219,000 measurements, a representative portion of the complete time series was taken. This was done by selecting a subsequence of 2,000 measurements that correspond to the middle of the layer, thus capturing different material hardness. Applying the GASF method

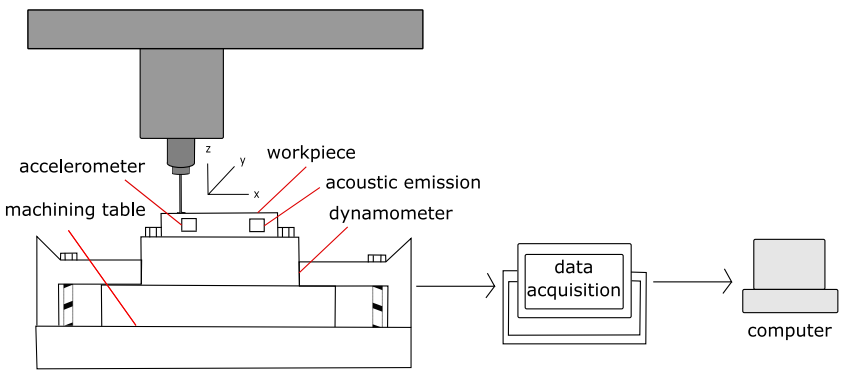


Fig. 7 Schematic of the experimental setup used in [54] to collect forces, vibrations and frequency stress waves of the cutting process

Table 1 Signal channels and measurement data of the complete dataset

Signal channel	Measurement data
Channel 1	F_x - cutting force in the X-dimension
Channel 2	F_y - cutting force in the Y-dimension
Channel 3	F_z - cutting force in the Z-dimension
Channel 4	V_x - vibration in the X-dimension
Channel 5	V_y - vibration in the Y-dimension
Channel 6	V_z - vibration in the Z-dimension
Channel 7	AE - acoustic emission

This study uses only those channels related to forces (top part of the table)

explained in Section 3, an image for each force (F_x, F_y and F_z) was obtained. These were then reduced from a size of $2k \times 2k$ pixels into images of 512×512 pixels using PAA and then combined into a 3-channel image. The associated wear class to this image is then determined by the flank wear value that was measured when the layer was removed. Although this experimental setup is particular to flank wear, the images could be labelled using other types of wear such as crater wear. Regardless of the type of wear measure used, the training process should be able to capture the features on the input that relate to the particular wear measure used.

As an example, Fig. 8 shows forces on the x-axis at different stages of the milling experiment. From what can be observed in this figure, the forces tend to be more uniform (i.e. shapes tend to get more circular) as the tool starts to wear out. The size reduction does not affect the time coherence of the data, allowing each individual flute temporal information to still be kept after PAA.

In total, the pre-processing step produced 315 3-channel images, one for each cutting event. This set of images was divided 70% for training and 30% for testing. The CNN was trained using the softmax regression method, which applies a softmax nonlinearity to the output of the network and calculates the cross-entropy between the normalised predictions and the actual labels. The parameters used for the training process are shown in Table 3.

Table 2 Operating conditions during dry milling

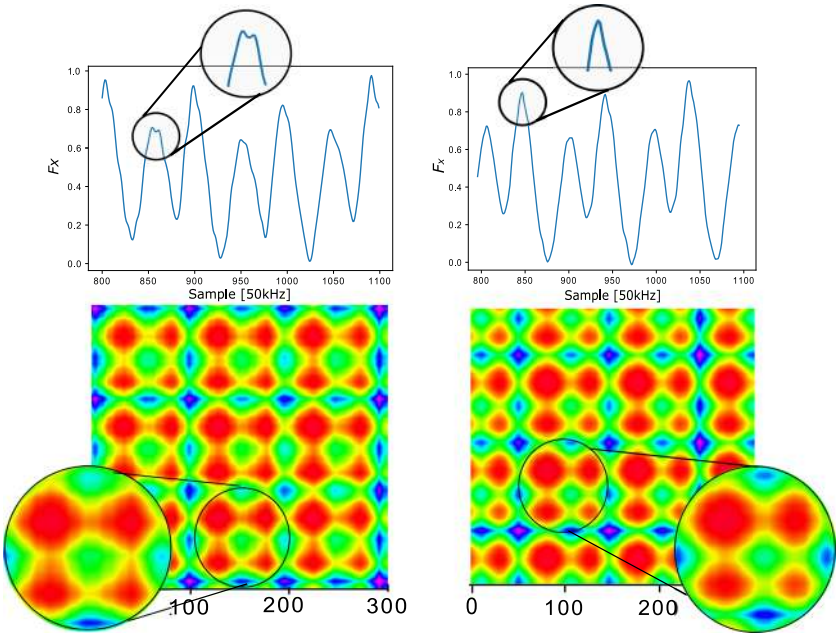
Parameter	Value
Spindle speed	10,400 RPM
Feed rate	1555 mm/min
Y depth of cut	0.125 mm
Z depth of cut	0.2 mm
Sampling rate	50 kHz/channel
Material	Stainless steel
Cutting tool	6 mm ball nose tungsten carbide cutter

Once the model was trained, it was tested on the remaining 95 images. Table 4 presents a confusion matrix with the results obtained. Based on the test set, the estimated accuracy of our model is 90%. Break-in wear was correctly classified for 82% of the cases, steady wear 94% of the cases and failure wear correctly classified 75% of the cases. The number of incorrect predictions suggest that the number of cases for break-in and failure regions may need to be increased.

As it can be observed in Fig. 8, the number of cuts that fall in the break-in region is 50, while the number of cuts in the steady-state are 200. This means that two-thirds of the data available would be categorised as steady-state. If the training set is generated by randomly sampling from the complete dataset, it is likely that two-thirds of those samples are steady-state class. This class imbalance problem has been well documented in the literature [55–57]. Failure cases tend to be considerably less abundant than steady wear cases. The less represented classes are more likely to be misclassified than the majority examples due to the design principles of the learning process. The training process optimises the overall classification accuracy which results in the misclassification of the minority classes. Therefore, several techniques could be applied to balance the number of samples of each class. Because the time series corresponding to one layer of the workpiece can be as long as 220,000 measurements, the data can be resampled. This would generate more than one sample from each layer, particularly with the break-in and failure cases. At the same time, an undersampling can be done by adding another class for the cases that are approaching the failure region. Thus, a fourth class that identifies this region could, in fact, be more useful as currently the low-wear region covers a wide range of tool wear values. It is important to remark that tool wear progresses differently depending on the type of tool, type of material, cutting parameters and other cutting conditions. It is not possible to identify the degree of class imbalance for a tool for which no prior data has been collected. Therefore, class imbalance needs to be detected and acted upon as part of the data preparation prior to model training.

A balanced number of cases among all classes will be crucial to achieve accuracy homogeneity across all wear regions. The overall results are nevertheless promising, showing that the CNN was successfully capable of capturing the intrinsic structures of the sensory data. This method is then scalable to include the remaining cut data.

A second experiment was performed by adding a 4th class that corresponds to the area prior to entering the failure region (Fig. 9). This area is of particular interest to this study as it considers a point in time were decisions could be taken to extend the life of the tool. The number of instances per case was also increased by taking two more sub-sequences from each layer, for a total of three 2,000 sample



(a) Flank wear V_b 98 mm labeled as Steady-state class

(b) Flank wear V_b 177mm labeled as Failure class

Fig. 8 Sample images of rescaled forces in the x-axis at different stages of flank wear. It can be observed how the shapes in the image become more circular as the signal becomes smoother. It can also be observed how the information by individual flute is kept

sub-sequences from the middle of each layer (cut event); enough so that the experiment could still be kept short for the proof of concept. Sequences were again encoded into images and labelled according to the wear value and the new classes. A total of 954 images were produced, where 70% was used for training and 30% for testing. The results are shown in Table 5.

The overall accuracy of the classification was 89%, which is about the same compared with the first experiment. However, there was an improvement on the percentage of cases correctly classified per class. For example, the break-in wear region went up from 82% in the previous experiment, whereas the steady wear region remains at 94%. The severe wear region, which was introduced in this round of experiments, is correctly classified 82% of the time. Despite this, it can be seen that only 6 cases (9%) of the severe region were classified as steady wear. The other 6 cases were classified as failure due to their proximity to the failure values. Finally, the failure region cases were accurately classified 82% of the time, which is again an improvement over the first experiment. From the number of cases, it can still be observed that there is a class imbalance that could be affecting the training process.

Table 3 O perating conditions during dry milling

Parameter	Value
Max steps	1000 steps
Learning rate	0.01
Learning rate decay factor	0.1
Number of examples per epoch	100 images
Number of epochs per decay	100 epochs
Training set size	220 images

In a third experiment, the class imbalance was addressed using a stratified undersampling technique. In the previous experiments, the datasets used for training were kept small to avoid high computational load for a proof of concept. However, it is possible to sample more subsequences from each of the 315 cuts. For the c_6 tool, it is possible to sample up to 95 subsequences from each cut, generating a total of 29,925 3-channel images. An undersampling strategy to deal with class imbalance is suitable in this case as the dataset is large enough to avoid losing critical features. Using a strata based on the wear classes defined, sampling of each class was done individually, making sure classes such as steady state were undersampled to achieve an equal number of samples across all classes. After performing the undersampling, a training set consisting of 14,000 images and a test set of 6,000 images were produced. These were used to train and validate a new model.

As the size of the training had increased considerably, images were reduced to 256 × 256. It was also decided to move from a generic Tensorflow architecture implementation to a more tuned one, by changing the size of the filters for both convolution layers from 5×5 to 16×16 for the first convolution and from 5×5 to 8×8 for the second convolution. Given that the GASF images are typically capturing 7 complete revolutions of the tool (21 cycles of the signal as the tool has 3 flutes), the kernel of the first convolution was set to a size of 16, which allows capturing a complete signal cycle. This means that the convolution will be searching for

Table 4 Confusion matrix summarising the results on the test set

$N_T = 95$	Actual break-in wear, $N(\%)$	Actual steady wear, $N(\%)$	Actual failure region, $N(\%)$
Predicted break-in wear	14 (82.35%)	2 (0.03%)	0 (0%)
Predicted steady wear	3 (17.64%)	62 (93.93%)	3 (0.25%)
Predicted failure region	0 (0%)	2 (0.03%)	9 (0.75%)
Total	17 (100%)	66 (100%)	12 (100%)

The table shows the classification given by the CNN for all the cases on the test set, indicating the number of correctly classified as well as the incorrectly classified. N_T refers to the total number of images in the test set

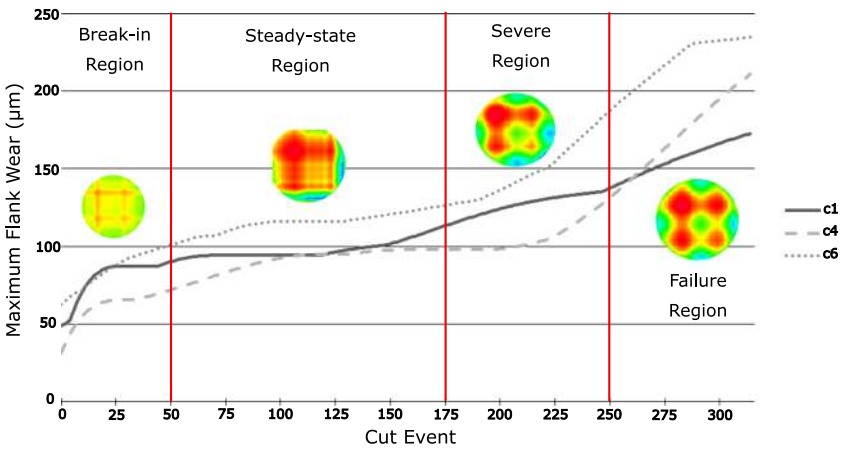


Fig. 9 Four stages of tool wear for cutters c_1, c_4 and c_6, and sample images of forces in the y-axis that correspond to those regions

Table 5 Confusion matrix summarising the results with four classes on the test set

$N_T = 282$	Actual break-in	Actual steady	Actual severe	Actual failure
	Wear $N(\%)$	Wear $N(\%)$	Wear $N(\%)$	$N(\%)$
Predicted break-in wear	32 (88.88%)	2 (1.5%)	0 (0%)	0 (0%)
Predicted steady wear	4 (11.11%)	125 (94%)	6 (8.69%)	0 (0%)
Predicted severe wear	0 (0%)	6 (4.5%)	57 (82.60%)	8 (18.18%)
Predicted failure region	0 (0%)	0 (0%)	6 (8.69%)	36 (81.81%)
Total	36 (100%)	133 (100%)	69 (100%)	44 (100%)

The table shows the classification given by the CNN for all the cases on the test set, indicating the number of correctly classified as well as the incorrectly classified. N_T refers to the total number of images in the test set

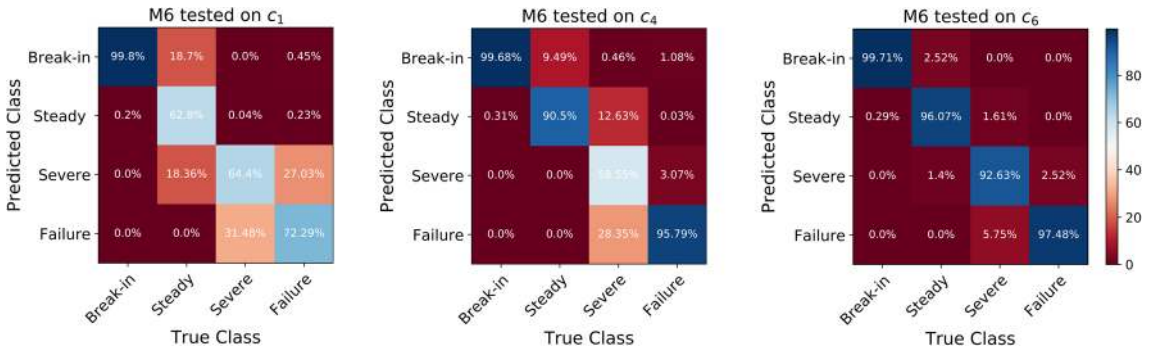


Fig. 10 Confusion matrices summarising the results of the $M6$ model (cutter c_6) with four classes using the stratified undersampling technique

features at a flute level. The stride of the kernel was set to 4 due to the size of the image, allowing a reduction of the feature map by a quarter. The pooling layer that follows uses a kernel of size 3, which allows a further reduction of the feature map to a size of 32×32. This is enough to keep the detected low-level features that will be grouped into higher level ones by the following convolution.

Results with the new trained model are shown in Fig. 10, where the model is labelled as $M6$, as it is the model that corresponds to cutter c_6. Overall, $M6$ was able to achieve a 96.4% accuracy on the test set. The classification accuracy increased for both the break-in and failure regions to 99.7% and 97.5% respectively when tested on c_6. The lowest accuracy was shown in the severe region, where a result of 92.6% correctly classified cases was achieved.

To understand the capabilities and limitations of the approach when a different set of data is available, a similar sampling and training was done with cutters c_1 and c_4, generating two additional models, $M1$ and $M4$, respectively. Each of these models were validated against the same cutter as well as the other two cutters. Accuracy results per class are shown in Fig. 11 and the overall results in Table 6. All experiments were carried out on a 2.80 GHz Intel Core i7-7600C CPU and 32GB RAM. The average training time for one batch (100 images) is 7.6 s, so a complete epoch takes approximately 16.5 min for any model. The testing time for one sample using any model is 0.2727 s. Although the training time is computationally expensive, testing is not, which still makes it applicable for real-time monitoring. Training time can be improved by using a higher specification processor or GPU as well as by parallelising the code and/or training one-class classifiers in parallel.

Table 6 Summary of the accuracy (in %) of each model (labelled $M1$, $M4$ and $M6$) when validated against the same cutter and other cutters

		Cutter		
		c_1 (%)	c_4 (%)	c_6 (%)
Model	$M1$	96	89.3	80.4
	$M4$	79.6	96.8	80.9
	$M6$	71.6	85.18	96.4

As can be observed in Table 6, there is not one model so far that works best when validated against all cutters. However, the model developed with c_1 ($M1$) achieves the highest accuracy across the three models when validated against other cutters (accuracy of 89.3% on c_4, and an accuracy of 80.4% on c_6). $M1$ particularly struggles classifying correctly the failure cases of c_6 (see Fig. 11 first row). Looking at Fig. 9, it can be seen that c_1 wears out at a very high rate during the first 20 cuts, reaching the steady state earlier than the other two cutters, and developing a lower tool wear after 315 cuts. This can explain why a model developed with this tool might perform badly on highly worn cutters as it does not provide enough examples of the

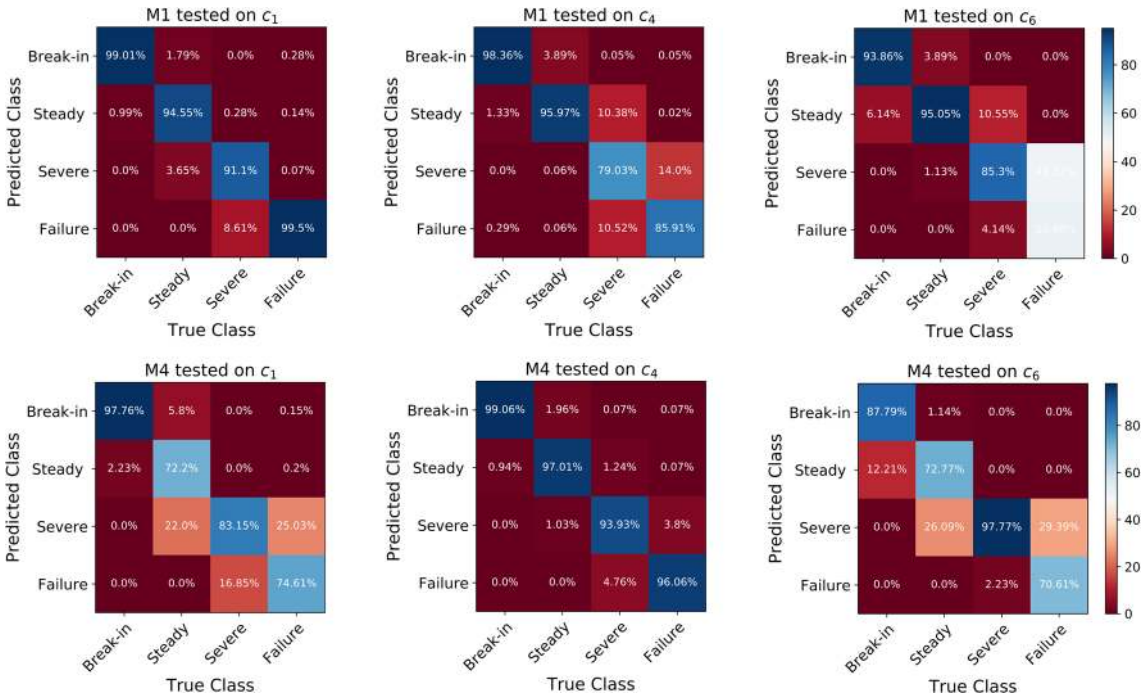


Fig. 11 Confusion matrices summarising the accuracy results (0–100 %) for $M1$ (top row) and $M4$ (bottom row) across c_1, c_4 and c_6 using four tool wear classes and the stratified undersampling for training/testing

degree of wear that was developed by c_6. Unfortunately, a more in-depth analysis onto these differences in wear degradation cannot be performed as no additional data or meta-data is available regarding the conditions of the PHM2010 data experiments. However, these results suggest that a better model can be built if a combination of both cutters' data was used in the training process.

When analysing the results obtained with $M6$, it was observed that this model is very good at identifying failure cases when tested on c_4. The model correctly classifies tool failure 95.8% of the time. This model shows again a weakness in identifying the severe region (see Fig. 10). Most of the cases that are incorrectly classified are identified as failure cases, which could be explained by the abrupt change in wear rate of c_4 when approaching failure. $M4$ did not show particularly good results when identifying tool failure. This model achieved 75% and 70% accuracy when tested in c_1 and c_6. What is interesting to point out is that $M4$ is particularly good at identifying the severe region on c_6, achieving a 97.7% accuracy. This again highlights the importance of making sure that a training dataset be a good representation of the search space in order to achieve generalisation.

In general, the results of the three models show the ability of the architecture used to learn force patterns and relate those to wear classes. The architectural setup of the CNN used in this last experiment allowed finding relevant features at a flute level, which is necessary for the approach to detect the current maximum wear regardless of the flute that is developing the wear. This is important, as it ensures that the technique can achieve good results regardless of the tool used. The accuracy obtained in particular classes shows the importance of presenting the CNN with samples that are representative of all the input space during training. A more robust model would need to be enriched with data from different cutters to ensure this.

5 Comparison of the proposed approach to previous work

The proposed approach has its advantages and disadvantages when compared with other approaches. Making a fair comparison in terms of accuracy is not straightforward due to several factors. First, to compare against classical machine learning, the best set of features would need to be found and not chosen arbitrarily. There are a wide range of algorithms for selecting and fusing features [58]; however, it is not in the scope of this paper to explore these. In addition, each approach has an "ideal" parametrisation depending on the problem and specific instantiation of the methodology, for example, selecting the right number of hidden layers and nodes in each layer of an ANN. For this reason, the comparison is approached differently, by describing the power of using GASF as a tool to automatically encode raw signals into images. The features of GASF images are ultimately exploited by an off-the-shelf CNN implementation that outputs the different stages of wear.

Most of the published works in tool wear prediction or tool wear classification perform some type of specific data pre-processing such as statistical feature selection using mean, maximum, standard deviation and median. Wu et al., for example, use these four features across multiple sensor data to perform tool wear prediction using ANNs, SVMs and random forests, the latter achieving the lowest root mean square error (RMSE) [9]. In Zhao et al., a deep learning approach using convolutional bi-directional LSTM (CBLSTM) network to perform tool wear prediction is presented. In this work, sensor signals are reduced from 200,000 measurements into 100 datums of maximum and mean values, and these are fed into the CBLSTM model. From three different configurations of the approach, the authors report that CBLSTM with dropout achieves the lowest RSME. [42]. The main disadvantage of manual feature extraction is that, unless it is continuously re-applied to update the models, it does not consider changes in the data distribution related to either noise or the tool wear phenomenon itself, making it unreliable in some cases. An example of this can be seen in cutter c_1. Inspecting the data of this cutter, it was found that, although mean, maximum and median statistics follow generally the same trend with a tendency to increase with every cutting event, there is a peculiar change in these statistics for cutter c_1 as seen in Fig. 12. The figure shows how there is a sudden increase in the maximum force along the x-axis (also applies for the mean, median and standard deviation) around cutting events 225 and 250, then the values return to their normal trend. Although change was not much in the wear measurements

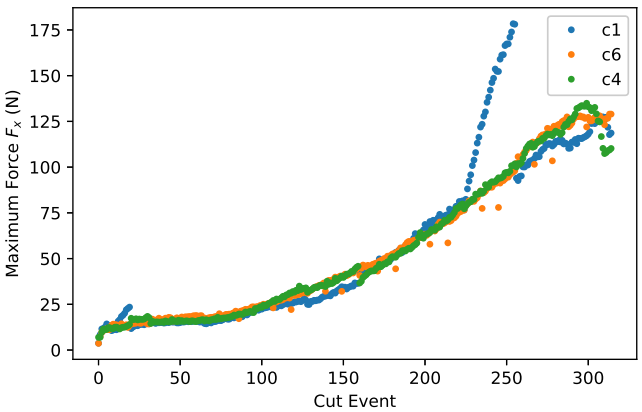


Fig. 12 Maximum force in newtons (N) in the x-axis at each cutting event for cutters c_1, c_4 and c_6

during this period of time (from 131.25 to 136.9 mm), the force values did show changes. This suggests that some conditions of the experiment changed and were reflected on the sensor readings but were not actually related to changes in tool wear. From the results reported in Zhao et al., it is interesting to note that the highest RMSE obtained is on cutter c_1, particularly during cutting events 225 and 250. This strongly suggests that there is a sensitivity to maximum and mean values, as the highest errors occur during the aforementioned cutting events. Although the method in [42] employs a deep learning approach, their results suggest that the model is not picking up the information on how one measurement changes in relation to another one in the time series, like a typical deep neural network would do. In their work, the dimensionality reduction performed averages 2k measurements, corresponding to nearly 7 revolutions of the tool, therefore losing the details of each individual flute. As each flute might wear out at a different rate, retaining flute level information is relevant as it provides a better understanding of how the tool is wearing through time. Figure 13 shows two force samples and their corresponding GASF images between cutting events 225 and 250. By visually inspecting the images, it can be inferred that not much change in the force patterns has happened during these cutting events. The GASF image encoding provides the CNN the right level of information for it to learn how the tool erodes at the flute level as well as how patterns change from one flute to another regardless of the actual force measurement made. From the results shown in Fig. 11, it can be seen that M4 achieves an accuracy of 83% on the severe cases of c_1. Taking into account that a third of the mean force measurements are showing a significant increase (Fig. 12), the CNN is still quite reliable in classifying these as severe (having only 17% as failure).

A similar comparison with the work of Wu et al. [9] is not straightforward as results are presented as total accuracy on the test set, with no detail of which tools were used for training and for testing. As a result, it is not possible to determine from the reported results how the proposed approaches are capable of dealing with the noise or changes in the data distribution.

A current disadvantage of the GASF representation is the loss of the magnitude information of the measurement during normalisation, as this normalisation process is performed individually by image, not taking into account the maximum value of all the observations. A combination of GASF and actual magnitude encoding could potentially be more effective, particularly for the cases like in c_1, where conditions could change suddenly.

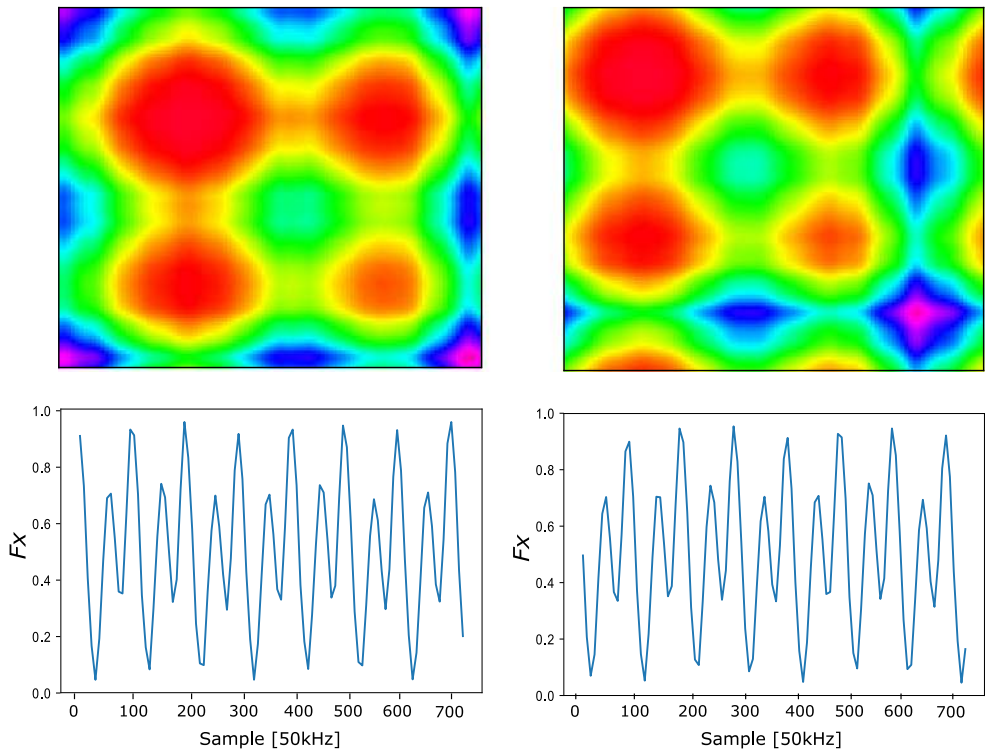


Fig. 13 Sample images of rescaled forces in the x-axis during cutting events: **a** 225 and **b** 250. Although there is a sudden increase on the mean force during cutting events 225 and 250 (which is not visible after normalisation), the wear does not increase at that same rate. In fact, the GASF images suggest there is not much change on the wear as the force patterns are very similar

(a) GASF image and rescaled forces F_x captured that correspond to a flank wear VB 131.47mm

(b) GASF image and rescaled forces F_x captured that correspond to a flank wear VB 135.62mm

6 Conclusions and future work

This paper presents an approach to tool wear classification by means of sensory data imaging and deep learning. The GASF encoding keeps the temporal correlations for each flute, which is an advantage over classification methods that are based on statistical features, where the features of a particular flute are lost. Experimental results show the ability of the CNN to capture and learn the features on the raw data to correctly classify tool wear condition. Overall, the percentage of accurately classified cases on the test set is high, achieving in most cases above 80% when testing in a new cutter. The moment prior to the transition from critical wear to failure is in most cases correctly identified, and the cases where it is incorrectly classified were generally labelled as a failure, which from an application standpoint means the replacement of the tool would still be enacted. These results show the importance of using a training sample set that can represent all of the input space. In this case, the training set needs to be enriched with samples from multiple cutters to ensure the successful detection of the transition period from severe to failure. The application of this work will allow for the extension of the remaining useful life of the tool, improve cut quality and ensure machining elements are replaced before failure.

Future work will include parallelisation of the architecture and its implementation to run in GPUs as well as incorporating the approach in a cloud architecture. Techniques for partially retraining the architecture will also be explored to study its adaptation capabilities when new data becomes available. Additional work will also include experimentation with more input channels on the GASF image to feed in multiple sensor data and improve the accuracy of the classification. Finally, further enhancements to the encoding technique will be investigated such as incorporating the magnitude information.

Funding information The authors received support from the Horizon 2020 MC-SUITE (ICT Powered MaChining Suite) project funded by the European Commission under grant agreement No 680478.

References

1. MacDougall W (2014) Industrie 4.0 Smart Manufacturing for the Future. GTAI Germany Trade and Invest
2. Wang J, Ma Y, Zhang L, Gao RX, Wu D (2018) Deep learning for smart manufacturing: Methods and applications. J Manuf Syst 48:144–156. https://doi.org/10.1016/j.jmsy.2018.01.003, special Issue on Smart Manufacturing
3. Bonifacio M, Diniz A (1994) Correlating tool wear, tool life, surface roughness and tool vibration in finish turning with coated carbide tools. Wear 173(1):137–144. https://doi.org/10.1016/0043-1648(94)90266-6
4. Ambhore N, Kamble D, Chinchanikar S, Wayal V (2015) Tool condition monitoring system: A review. Mater Today: Proc 2(4):3419–3428. https://doi.org/10.1016/j.matpr.2015.07.317, 4th International Conference on Materials Processing and Characterzation
5. Kong D, Chen Y, Li N (2017) Force-based tool wear estimation for milling process using gaussian mixture hidden markov models. Int J Adv Manuf Technol 92(5):2853–2865. https://doi.org/10.1007/s00170-017-0367-1
6. Niaki FA, Ulutan D, Mears L (2015) In-process tool flank wear estimation in machining gamma-prime strengthened alloys using kalman filter. Procedia Manuf 1:696–707. https://doi.org/10.1016/j.promfg.2015.09.018, 43rd North American Manufacturing Research Conference, NAMRC 43, 8-12 June 2015, UNC Charlotte, North Carolina, United States
7. Wang P, Gao RX (2015) Adaptive resampling-based particle filtering for tool life prediction. J Manuf Syst 37:528–534. https://doi.org/10.1016/j.jmsy.2015.04.006
8. Cosme LB, D'Angelo MFSV, Caminhas WM, Yin S, Palhares RM (2018) A novel fault prognostic approach based on particle filters and differential evolution. Appl Intell 48(4):834–853. https://doi.org/10.1007/s10489-017-1013-1
9. Wu D, Jennings C, Terpenny J, Kumara S (2016) Cloud-based machine learning for predictive analytics: Tool wear prediction in milling. In: 2016 IEEE International Conference on Big Data (Big Data), pp 2062–2069. https://doi.org/10.1109/BigData.2016.7840831
10. Sick B (2002) On-line and indirect tool wear monitoring in turning with artificial neural networks: a review of more than a decade of research. Mech Syst Signal Process 16(4):487–546. https://doi.org/10.1006/mssp.2001.1460
11. Wuest T, Weimer D, Irgens C, Thoben KD (2016) Machine learning in manufacturing: advantages, challenges, and applications. Prod Manuf Res 4(1):23–45. https://doi.org/10.1080/21693277.2016.1192517
12. Terrazas G, Martínez-Arellano G, Benardos P, Ratchev S (2018) Online tool wear classification during dry machining using real time cutting force measurements and a cnn approach. J Manuf Mater Process 2(4):72. https://doi.org/10.3390/jmmp2040072
13. PHMSociety (2010) 2010 phm society conference data challenge, https://www.phmsociety.org/competition/phm/10, Accessed January 31, 2018
14. Cui X, Zhao J, Dong Y (2013) The effects of cutting parameters on tool life and wear mechanisms of cbn tool in high-speed face milling of hardened steel. Int J Adv Manuf Technol 66(5):955–964. https://doi.org/10.1007/s00170-012-4380-0
15. Taylor F (1907) On the art of cutting metals. Trans Am Soc Mech Eng 38:31–35
16. Poulachon G, Moisan A, Jawahir I (2001) Tool-wear mechanisms in hard turning with polycrystalline cubic boron nitride tools. Wear 250(1):576–586. https://doi.org/10.1016/S0043-1648(01)00609-3, 13th International Conference on Wear of Materials
17. Karandikar JM, Abbas AE, Schmitz TL (2013) Tool life prediction using random walk bayesian updating. Mach Sci Technol 17(3):410–442. https://doi.org/10.1080/10910344.2013.806103
18. Sun J, Rahman M, Wong Y, Hong G (2004) Multiclassification of tool wear with support vector machine by manufacturing loss consideration. Int J Mach Tools Manuf 44(11):1179–1187. https://doi.org/10.1016/j.ijmachtools.2004.04.003

19. Özel T, Karpat Y (2005) Predictive modeling of surface roughness and tool wear in hard turning using regression and neural networks. Int J Mach Tools Manuf 45(4):467–479. https://doi.org/10.1016/j.ijmachtools.2004.09.007
20. Palanisamy P, Rajendran I, Shanmugasundaram S (2008) Prediction of tool wear using regression and ANN models in end-milling operation. Int J Adv Manuf Technol 37:29–41. https://doi.org/10.1007/s00170-007-0948-5
21. Sanjay C, Neema M, Chin C (2005) Modeling of tool wear in drilling by statistical analysis and artificial neural network. J Mater Process Technol 170(3):494–500. https://doi.org/10.1016/j.jmatprotec.2005.04.072
22. Chungchoo C, Saini D (2002) On-line tool wear estimation in cnc turning operations using fuzzy neural network model. Int J Mach Tools Manuf 42(1):29–40. https://doi.org/10.1016/S0890-6955(01)00096-7
23. Ferry N, Terrazas G, Kalweit P, Solberg A, Ratchev S, Weinelt D (2017) Towards a big data platform for managing machine generated data in the cloud. In: 2017 IEEE 15th International Conference on Industrial Informatics (INDIN), pp 263–270. https://doi.org/10.1109/INDIN.2017.8104782
24. Liao TW (2005) Clustering of time series data - a survey. Pattern Recogn 38(11):1857–1874
25. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Proceedings of Advances in neural information processing systems, curran associates inc., USA, NIPS'12, vol 25, pp 1090–1098
26. Ciresan DC, Meier U, Schmidhuber J (2012). Multi-column deep neural networks for image classification. arXiv:1202.2745
27. Hinton G, Deng L, Yu D, Dahl GE, Mohamed AR, Jaitly N, Senior A, Vanhoucke V, Nguyen P, Sainath TN, Kingsbury B (2012) Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Proc Mag 29(6):82–97. https://doi.org/10.1109/MSP.2012.2205597
28. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks, pp 3104–3112. http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf
29. LeCun Y, Bengio Y, Hinton GE (2015) Deep learning. Nature 521(7553):436–444. https://doi.org/10.1038/nature14539
30. Malhi A, Yan R, Gao RX (2011) Prognosis of defect propagation based on recurrent neural networks. IEEE Trans Instrum Meas 60(3):703–711. https://doi.org/10.1109/TIM.2010.2078296
31. Zhao R, Wang D, Yan R, Mao K, Shen F, Wang J (2018) Machine health monitoring using local feature-based gated recurrent unit networks. IEEE Trans Ind Electron 65(2):1539–1548. https://doi.org/10.1109/TIE.2017.2733438
32. LeCun Y, Bengio Y (1998) The handbook of brain theory and neural networks. MIT Press, Cambridge. Convolutional Networks for Images, Speech, and Time Series, pp 255–258
33. Chen X, Xiang S, Liu C, Pan C (2014) Vehicle detection in satellite images by hybrid deep convolutional neural networks. IEEE Geosci Remote Sens Lett 11(10):1797–1801. https://doi.org/10.1109/LGRS.2014.2309695
34. Dieleman S, Willett KW, Dambre J (2015) Rotation-invariant convolutional neural networks for galaxy morphology prediction. Mon Not R Astron Soc 450(2):1441–1459. https://doi.org/10.1093/mnras/stv632
35. Pereira S, Pinto A, Alves V, Silva CA (2016) Brain tumor segmentation using convolutional neural networks in mri images. IEEE Trans Med Imaging 35(5):1240–1251. https://doi.org/10.1109/TMI.2016.2538465
36. Kiranyaz S, Ince T, Gabbouj M (2016) Real-time patient-specific ecg classification by 1-d convolutional neural networks. IEEE Trans Biomed Eng 63(3):664–675. https://doi.org/10.1109/TBME.2015.2468589
37. Zheng Y, Liu Q, Chen E, Ge Y, Zhao JL (2014) Time series classification using multi-channels deep convolutional neural networks. In: Li F, Li G, Sw H, Yao B, Zhang Z (eds) Web-age information management. Springer International Publishing, Cham, pp 298-310
38. Levent E (2017) Bearing fault detection by one-dimensional convolutional neural networks. Math Probl Eng 2017
39. Li S, Liu G, Tang X, Lu J, Hu J (2017) An ensemble deep convolutional neural network model with improved d-s evidence fusion for bearing fault diagnosis. Sensors 17(8). https://doi.org/10.3390/s17081729
40. Zhang W, Peng G, Li C (2017) Bearings fault diagnosis based on convolutional neural networks with 2-d representation of vibration signals as input. In: MATEC Web of conferences, EDP sciences, vol 95, pp 13001
41. Abdeljaber O, Avci O, Kiranyaz S, Gabbouj M, Inman DJ (2017) Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. J Sound Vibr 388:154–170. https://doi.org/10.1016/j.jsv.2016.10.043
42. Zhao R, Yan R, Wang J, Mao K (2017) Learning to monitor machine health with Convolutional Bi-Directional LSTM Networks. Sensors 17(2):273. https://doi.org/10.3390/s17020273
43. Wang Z, Oates T (2015) Imaging time-series to improve classification and imputation. In: Proceedings of the 24th International Conference on Artificial Intelligence, IJCAI'15. AAAI Press, pp 3939–3945
44. Eckmann JP, Kamphorst SO, Ruelle D (1987) Recurrence plots of dynamical systems. Europhys Lett (EPL) 4(9):973–977. https://doi.org/10.1209/0295-5075/4/9/004
45. Silva DF, Souza VMAD, Batista GEAPA (2013) Time series classification using compression distance of recurrence plots. In: 2013 IEEE 13th International Conference on Data Mining, pp 687–696. https://doi.org/10.1109/ICDM.2013.128
46. Donner RV, Small M, Donges JF, Zou Y et al (2011) Recurrence-based time series analysis by means of complex network methods. Int J Bifurcat Chaos 21(04):1019–1046
47. Campanharo ASLO, Sirer MI, Malmgren RD, Ramos FM, Amaral LAN (2011) Duality between time series and networks. PLOS ONE 6(8):1–13. https://doi.org/10.1371/journal.pone.0023378
48. Chen J, Chen W, Huang C, Huang S, Chen A (2016) Financial time-series data analysis using deep convolutional neural networks. In: 2016 7th International Conference on Cloud Computing and Big Data (CCBD), pp 87–92. https://doi.org/10.1109/CCBD.2016.027
49. Keogh EJ, Pazzani MJ (2000) Scaling up dynamic time warping for datamining applications. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '00. ACM, New York, pp 285–289. https://doi.org/10.1145/347090.347153
50. Groover M (2010) Fundamentals of modern manufacturing, Materials, Processes, and Systems. Wiley, New York
51. Binder M, Klocke F, Doebbeler B (2017) An advanced numerical approach on tool wear simulation for tool and process design in metal cutting. Simul Modell Pract Theory 70:65–82. https://doi.org/10.1016/j.simpat.2016.09.001
52. Géron A (2017) Hands-On Machine learning with Scikit-Learn and tensorflow: Concepts, Tools and Techniques to build Intelligent Systems. O'Reilly Media Inc
53. Tensorflow (2017) Convolutional neural networks, https://www.tensorflow.org/tutorials/deep_cnn, Accessed January 23, 2018
54. Li X, Lim BS, Zhou JH, Huang SC, Phua S, Shaw KC (2009) Fuzzy neural network modelling for tool wear estimation in dry

milling operation. In: Annual Conference of the Prognostics and Health Management Society, pp 1–11
55. Longadge R, Dongre S (2013) Class imbalance problem in data mining review. arXiv:1305.1707
56. Khan SH, Hayat M, Bennamoun M, Sohel FA, Togneri R (2018) Cost-sensitive learning of deep feature representations from imbalanced data. IEEE Trans Neural Netw Learn Syst 29(8):3573–3587. https://doi.org/10.1109/TNNLS.2017.2732482
57. Lee T, Lee KB, Kim CO (2016) Performance of machine learning algorithms for class-imbalanced process fault detection problems. IEEE Trans Semicond Manuf 29(4):436–445. https://doi.org/10.1109/TSM.2016.2602226
58. Wang J, Xie J, Zhao R, Zhang L, Duan L (2017) Multisensory fusion based virtual tool wear sensing for ubiquitous manufacturing. Robot Comput-Integr Manuf 45:47–58. https://doi.org/10.1016/j.rcim.2016.05.010, special Issue on Ubiquitous Manufacturing (UbiM)