

Investigation on the influence of tool path radius on tool wear and tool life prediction in micro-milling using neural networks

Sumant Bagri¹, Ashish Manwar¹, Alwin Varghese¹, Soham Mujumdar¹, Suhas S. Joshi¹¹

Department of Mechanical Engineering, Indian Institute of Technology Bombay.

Abstract

Since, the tool size used in micro-milling is in the micrometer range, the stability as well as the part quality strongly depend on the forces and vibrations acting on the tool. It is therefore important to monitor the micro-milling process to ensure system stability and maintain production standards and in this context monitoring and prediction system have been developed based on machine learning models. Various methods have been implemented in the past to capture the sudden failure of microtools by correlating the failure with varying process parameters. However the impact of non-linear tool paths on tool wear is not accommodated in most of these studies. The main aim of this project is to investigate the impact of semi-circular tool paths on the sudden tool failure by introducing a new process parameter called the tool path radius. An experimental setup was designed to measure sensory data such as the cutting forces, acceleration and vibration as well as the tool wear to train the neural-network models. Sensor data was segmented into individual slot data using k-means clustering and peaks-analysis algorithms and was subsequently used for statistical feature extraction. Image binarization and alignment operations were performed on the tool condition images to extract wear features such as diameter reduction as well as the wear of individual cutting edges. Wear state classification criteria were defined to categorize the tool condition into three regions of initial wear, steady-state wear and critical wear. In order to better understand the extremely complex mechanism of microtool failure, artificial neural networks as well as deep belief networks were implemented to predict the wear state as well as the remaining-useful-life of the tools. The models were trained using hyperparameter tuning techniques to improve prediction accuracy. It was found that the wear rate increased with increasing tool path radius and larger values were found to cause catastrophic failure. The major wear mechanisms were

also found to vary with varying tool path radius. The neural-network models gave 93-98% accuracy for wear state classification and RUL prediction.

Keywords: Micro end-milling, Tool path radius, Deep Neural Networks

1. Introduction

Miniaturised components demand higher dimensional conformity and surface finish quality as compared to macro-sized parts. The micro-parts produced often have a very complex geometry involving multiple curvatures which require non-linear tool paths having first-order discontinuities. These will necessitate tool slowdowns and stoppages leading to unwanted fluctuations in feed rate and jerks, resulting in inconsistent machining, poor surface qualities and even tool breakage [1]. Tool condition monitoring (TCM) has been recognized as an important process in micro-milling operations to prevent excessive tool wear and to maintain part tolerances and surface quality [2]. It involves gathering multi-sensor data, such as forces, vibrations, acoustic emissions, along with capturing tool condition using optical and data-driven techniques, to generate reference models for condition and performance monitoring.

Data-driven methods based on machine learning algorithms as well as probabilistic models have been increasingly used for TCM in the past decade. Malekian et al. [2] used accelerometer, force and acoustic emission sensors to develop a sensor fusion based neuro-fuzzy model which determines the tool wear in micro-milling. Similarly, Li et al. [3] used force, accelerometer and AE sensors to develop a Fuzzy Neural Network (FNN) to predict the cutter flank wear in dry milling. They reported superiority of FNN, in predicting tool wear, over other multi-regression models and backpropogated neural-networks. Cho et al. [4] used sensory signals to train decision fusion models including support vector machine, multilayer perceptron neural network (MLP) and radial base function neural network (RGBF) for TCM. Their TCM design based on feature fusion significantly improved the accuracy of tool condition classification. Zhenhua Wu [5] used adaptive neuro-fuzzy inference system (ANFIS) along with support vector regression and neural network for estimating tool wear. The prediction of tool wear in milling using three popular machine learning algorithms, including ANNs, SVR, and RFs was done by Wu et al. [6]. Using the spindle power sensor data and ANN based regression model, Drouillet et

al. [7] predicted remaining-useful-life (RUL) for different spindle speeds in end-milling. In a similar study for gun-drilling process, Zhang et al. [8] implemented a three-hidden-layer multi layer perceptron, to predict RUL. Their study provided baseline performances for many benchmark models such as gradient boosting regressor (GB), Stochastic Gradient Descent Regressor (SGD), ElasticNet (EN) to name a few, for tool condition monitoring based on RUL prediction. Another comparative study of SVR and ANN with an emerging deep neural network methodology called the deep belief networks (DBNs) was done by Chen et al [9]. They reported the superiority of DBNs for predicting tool wear using multi-sensor data over the other two methodologies. Addressing the issue of deploying TCM systems for industrial purposes, Li et al. [10] developed an audio signals based monitoring system using classification models such as CART, random forest, kNN and SVM.

The material removal processes during end-milling along circular and linear tool paths have been generally assumed to be similar [1]. The cutting tooth trajectory is epitrochoidal in circular tool paths as opposed to the circular/trochoidal tooth trajectory in linear tool paths. Thus, the material removed in each tooth pass would be different in the two tool path geometries. Consequently, the cutting force/vibration profiles produced would be different. Banerjee and Bordatchev [1], investigating the impact of tool path geometry, reported that a smaller tool path radii results in lower cutting force amplitudes whereas, a larger tool path radii, equal to 50 mm, have cutting forces similar to those along the linear tool path. In a similar study, Ma et al. [11], in their work on high speed milling of Ti-6Al-4V, found that the cutting forces and vibrations increase with an increase in the tool path radius. Silva and Hassui [12] also conducted an experimental investigation using linear, sinusoidal, triangular and sawtooth tool paths. They reported a reduction in cutting force magnitudes when using sinusoidal (45-70%), triangular (45–70%) and sawtooth (30–35%) tool paths in comparison to the linear path. Shi et al. [13] reported that smaller tool path radii lead to greater uncut chip thickness and that adjusting feed direction can minimize cutting forces.

Although, a substantial amount of work has been done to develop the most optimal, data-driven, TCM system for micro-milling using sensory data and process parameters,

tool path geometry remains unaccounted for. It is evident from the above review that there is a need to specifically consider the effect of circular tool paths in end-milling. The main aim of this research is to study and model the impact of circular tool paths on the tool wear and tool life in micro-end-milling using neural network based data-driven techniques. The focus here is to bridge the gap between studying the effect of tool path radius on cutting forces/vibrations and using such sensory data for a data-driven tool condition monitoring. With the goal to better understand the tool wear and to improve prediction prognostics, the wear of individual cutter edges was also separately considered. The modelling methodology explores the newly developing domain of deep neural networks for data-driven prognostics to predict tool condition and remaining-useful-life (RUL). We envisage that such models would be able to capture the effect of tool path radius on the wear state and life of the tool for on-line tool condition monitoring.

The remainder of the paper is organized into four major sections: (i) Theme and methodology of work, (ii) Data Acquisition and Processing, (iii) Design of neural network architecture and training and (iv) Results and discussions followed by conclusions.

2. Experimental Details

2.1. Theme of Work

The theme of this work is summarized in the flow chart in Fig. 1.

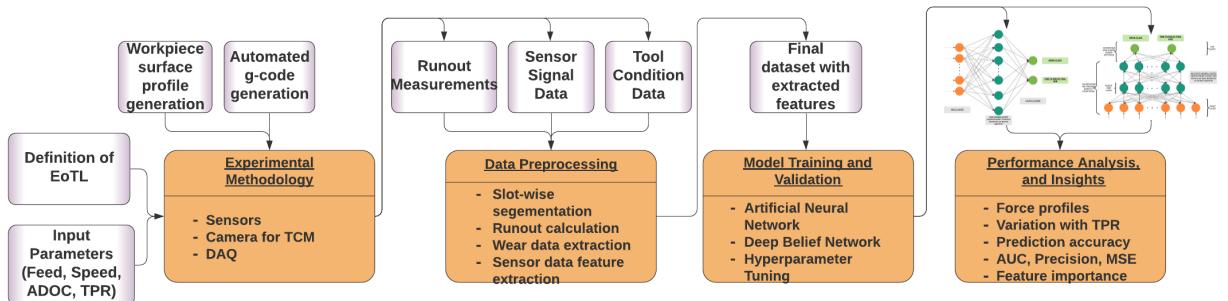


Figure 1: Project work flow

In order to understand the effect of tool-path on tool wear, curved slot milling operations were carried out with varying radius of semi-circular tool-paths as discussed in section 2.3 below. The experiments were carried out until the end-of-tool-life (EoTL) criteria was reached, which is defined by Eq. 1, or until tool breakage.

$$EoTL : \text{Diameter Reduction} \geq 150 \mu\text{m} \quad (1)$$

The process parameters that were varied include the axial depth of cut (ADoC), feed per tooth and rotational speed of cutter. In addition, tool path radius (tpr) was also varied. The data generated from the experiments was processed to extract statistical features and output variables for classification, training, validation and testing datasets. It was then fed to two neural-network models, listed in section 4, for training and evaluating prediction error on testing data using multiple metrics.

Two types of prediction analyses were performed; these include classification based training to predict wear category and regression-based training for prediction of RUL (remaining-useful-life). The classification based training involves training three separate models using different parameters as output labels namely: i) diameter reduction, ii) chipping area of edge 1 and iii) chipping area of edge 2 (tools with two flutes were used). The trained models were then combined to give an overall view of the tool condition. Such a combination of models to classify the wear state of each edge and the tool as a whole enables a more informed decision making process. The output from the regression based model would provide an insight into the remaining tool life.

2.2. Experimental Setup

The overview of the experimental setup with the signal amplifiers, DAQ console and laser vibrometer is shown the Fig. 2(a). Fig. 2(b) displays the workpiece setup mounted on a dynamometer along with a camera attached to the fixture, and an accelerometer placed on the spindle head. The machining center was connected to an external lubrication system to provide temperature control for high rpm spindle working conditions. The maximum speed that the spindle can reach is 140,000 RPM. The vertical linear motion in Z-direction is given to the spindle head and the X- and Y- directions are given to the table. The CNC controller is interfaced with the CNC-train software to provide programmable control over the micro-CNC machine. The zero-level in the Z-direction was setup using a camera interfaced with the DINOWARE software.

One of the major goals of the experiments was to be able to capture the tool condition without removing the tool using a high resolution camera provided by Dino-Lite. In order

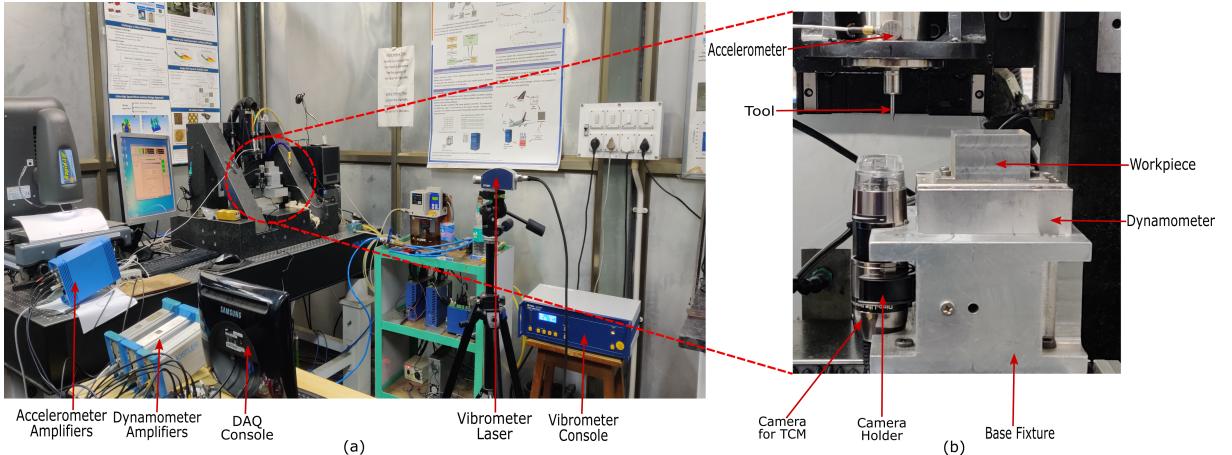


Figure 2: Complete layout of experiment (a) Overview of entire setup (b) Machining center

to achieve this, a fixture was specifically designed and fabricated to house the camera. The images of the tool face were captured at a predetermined interval as is discussed in section 2.3. The camera was connected to a separate system which was also used for the sensor data acquisition.

The workpiece used in the experiments was a plate of grade 304 stainless steel (SS304) and had dimensions that provide $35 \times 40 \text{ mm}^2$ machining area. It was mounted on the dynamometer using screws. The uncoated tungsten-carbide (WC) micro-end mills of shank diameter 3 mm and the tool diameter $500 \mu\text{m}$ were used in all the experiments (see Table 1 for tool specifications). The tool was fitted into a collet such that it maintains an overhang of 20 mm across all the experiments.

Table 1: Tool Specifications

Diameter	Flute Length	Helix Angle	Overall Length	Shank Diameter
$500 \mu\text{m}$	1 mm	30°	38 mm	3 mm

The cutting forces were measured using a Kistler, compact multi-component dynamometer while the acceleration data was captured with the help of Kistler IEPE Triaxial Accelerometer. The charge signals generated from the dynamometer and the accelerometer were amplified and converted to voltage signals. In order to capture more characteristics of the system under study, vibration measurement was done using a laser doppler vibrometer (Polytec NVL-500). Its controller had a built-in signal processor, power supply unit and a laser interferometer. It is a non-intrusive method that provides

signals of the tool vibration velocity.

2.3. Design of Experiments

The process parameters and their corresponding levels are listed in Table 2.

Table 2: Experimental process parameters and levels

Parameters	Level 1	Level 2	Level 3
Feed ($\mu\text{m}/\text{tooth}$)	4	6	8
Spindle speed (RPM)	40000	50000	60000
DOC (μm)	50	80	110
Tool Path Radius (μm)	250	500	750

In order to best accomodate for the variability in machining parameters, a design of experiment involving a 3-level, orthogonal array was used. Eight experiments were performed till the end of tool life using the parametric conditions given in Table 3.

Table 3: Final experimental process parameters

Experiment No.	Tool Id	Feed Rate (mm/s)	RPM	Depth of cut	Tool path radius
1	T79	16	60000	80	250
2	T80	10	50000	80	250
3	T81	5.33	40000	50	250
4	T82	10.67	40000	110	500
5	T83	12	60000	50	500
6	T84	6.67	50000	80	500
7	T85	8	60000	110	750
8	T86	13.33	50000	50	750

One of the key features of the experiments was generation of semi-circular tool paths with varying radius. A photograph of a typical workpiece after machining is shown in Fig. 3(a) and the tool path geometry for each slot is shown in Fig. 3(b). A circular tool path constantly changes the feed direction in a sinusoidal manner and is therefore expected to affect tool wear progression differently than a straight line path. The photographs of the

worn tools were taken at smaller or larger intervals depending upon whether the previous image of the tool indicated a severe or insignificant degradation, respectively.

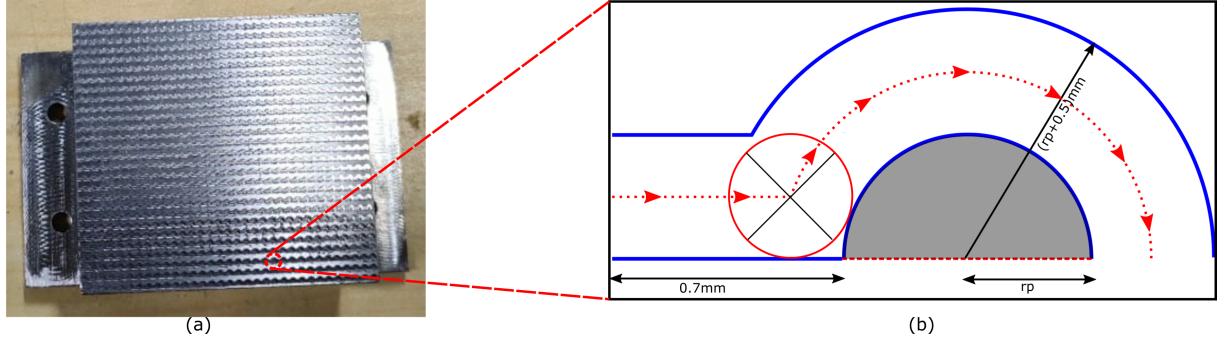


Figure 3: Tool path design (a) Workpiece after machining (b) Tool path geometry

2.4. Experimental Procedure

The main components of the experimental procedure are described in the flow chart displayed in Fig 4.

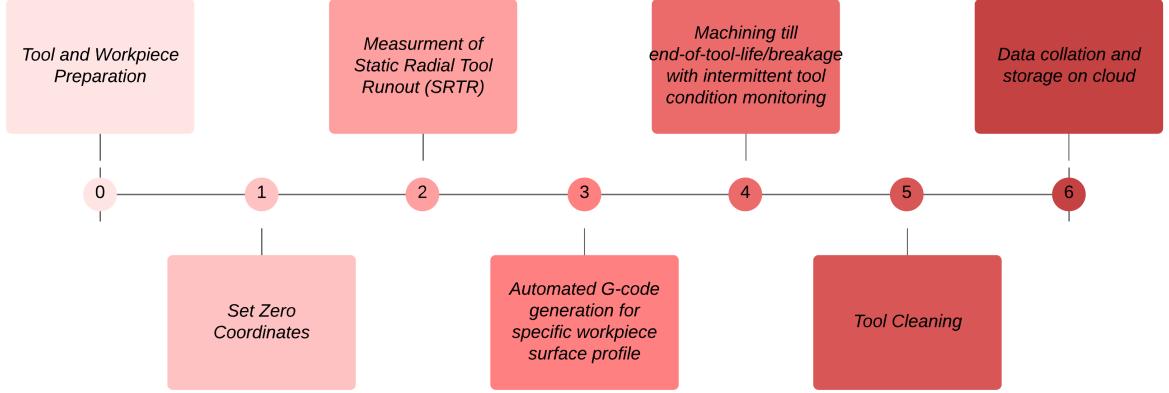


Figure 4: Flow for experimental protocol

2.4.1. Tool and workpiece preparation

One of the major goals was to reduce the exacerbation of tool wear due to varying static runout caused by repeated removal of the tool. By keeping the tool in-situ throughout each experiment ensured consistent static tool-runout which was measured before starting the experiments. The zero coordinates of the micro-CNC machine were then set using

a separate, low resolution camera. The initial image of the tool was then captured and saved while maintaining relative position of the two flutes with the help of a marking. This helped in tracking the wear of each flute while maintaining consistency. Subsequently the static radial tool runout of the tool was captured to be given as an input parameter to the machine learning model. This was achieved by taking tool images rotated at an interval of 30° . The static radial runout can be obtained by measuring the distance between the centers of a pair of images phase-shifted by 180° .

2.4.2. G-code Generation for Slot Milling

Since the machining strategy involved using dynamic batch sizes for number of slots machined, large number of G-code files were obtained by automating the process of G-code generation using a Python script. Fig. 5 shows the entire flow for generation of G-codes which is described as follows. The microtool was used as a coordinate measuring tool and was touched at different locations on the workpiece surface. The z-coordinates at these locations were recorded and passed to a multiple linear regression model. This model would be fit based on the input z-coordinates and subsequently be used as a mathematical representation of the workpiece surface. This representation along with the feed rate and tool path radius was passed to the python script which then generated necessary G-codes to machine the entire workpiece with variable slot numbers (batch sizes of 5-slots, 15-slots and 30-slots). The slots were then machined until the end-of-tool-life criteria was reached as described in Eq. 1 or till tool breakage.

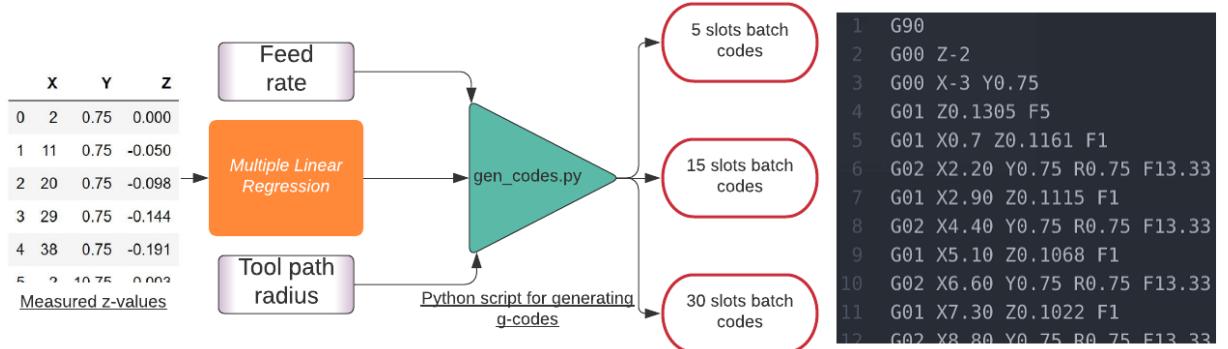


Figure 5: Automated G-code generation

2.4.3. Post-slot-milling

After machining each batch of slots, tools were cleaned and their condition was recorded using a high-resolution camera. The orientation of the tool was adjusted using a marker on the tool, to maintain the relative flute position while taking the images. The sensor data stream was also stopped at this point and stored in a separate file in the DAQ system. After each experiment is completed, the entire dataset was stored on the cloud for ease of accessibility.

3. Data Processing

After the data acquisition, preprocessing techniques were employed; see Fig. 6 for the data processing flow.

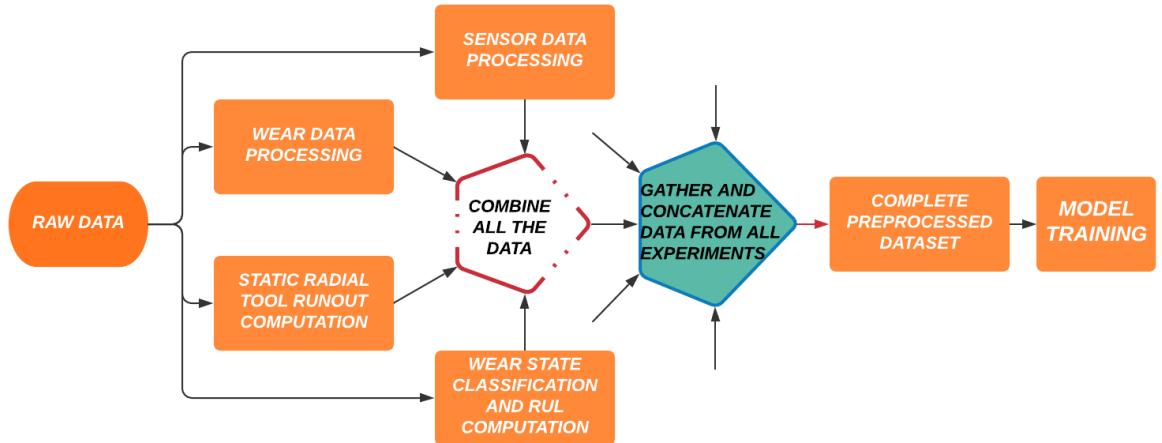


Figure 6: Data preprocessing flow

3.1. Sensor Data Preprocessing

The data files for all slot batches were imported into the working environment recursively, one at a time. The columns in the raw imported dataset were as listed: time, forces (x-, y-, z-directions), acceleration (x-, y-, z-directions) and vibration velocity. Each of the sensor data columns were multiplied by corresponding multiplication factors. Subsequently, the machining and non-machining data-points were identified using K-means clustering analysis with 10 clusters. K-means clustering is an unsupervised machine learning technique for clustering the dataset into k different groups. The red circles in Fig.

7 are the centroids of the 10 different clusters which indicate a further grouping of the data into two categories as shown. Inorder to further improve identification of start and endpoints of the machining and non-machining data, a peaks-analysis algorithm was implemented. This algorithm analyzes the variation in peak amplitudes of the data to detect cluster boundaries with a higher precision.

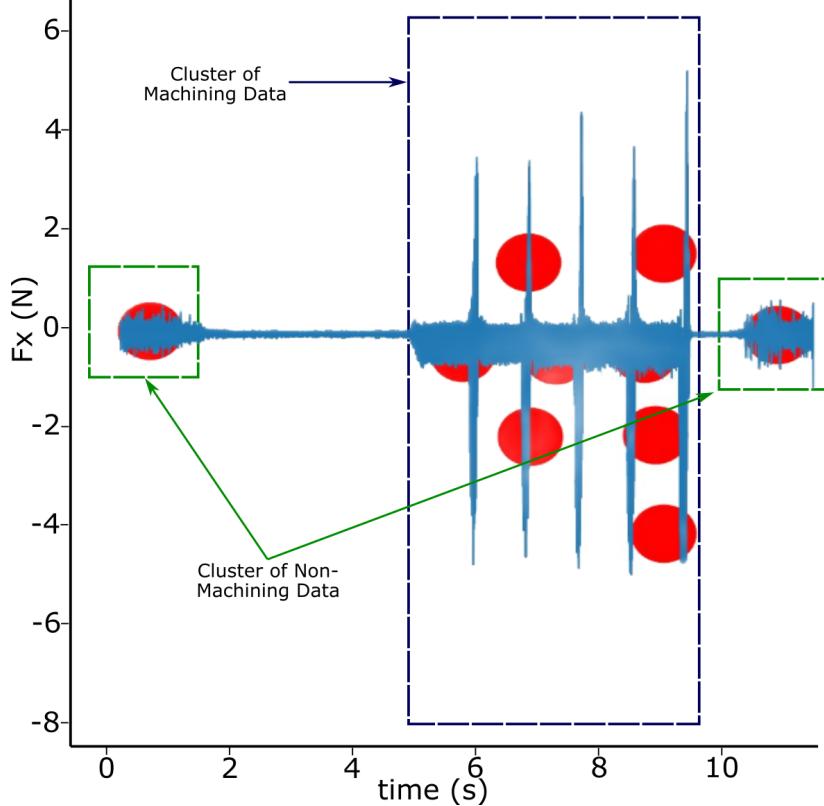


Figure 7: Machining and non-machining clustering

Using the peaks-analysis algorithm, the combined, semi-circular slot data was obtained from the machining data as shown in Fig. 8(a). This was then subjected to further segmentation into individual slot data, see Fig. 8(b). The fine-tuning of the individual slot data involved detecting and removing noise and trimming the slot data to a higher precision. The final slot-wise segmented data transformation is displayed in Fig. 8(c). Subsequently, each slot data was further segmented into 4 parts for statistical feature extraction as described in Table 4. At this point, the various process parameters: ADOC (axial depth-of-cut), feed, spindle speed and tpr (tool path radius), are also concatenated to the dataset.

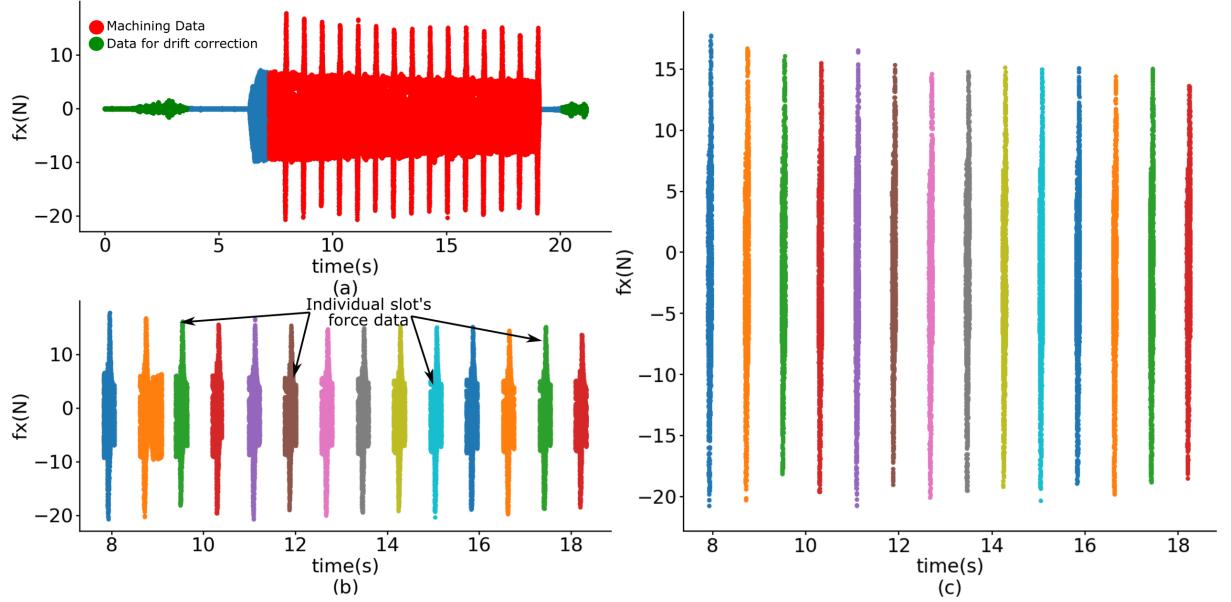


Figure 8: Slot segmentation: (a) Initial segmentation (b) Slot data detection (b) Fine-tuning detected slot data

Table 4: Feature extraction for sensor data

Fx	Fy	Fz	Vx	Vy	Vz
Mean	Mean	Mean	Mean	Mean	Mean
RMS	RMS	RMS	RMS	RMS	RMS
Std.Dev	Std.Dev	Std.Dev	Std.Dev	Std.Dev	Std.Dev
Max	Max	Max	Max	Max	Max
Min	Min	Min	Min	Min	Min
Kurtosis	Kurtosis	Kurtosis	Kurtosis	Kurtosis	Kurtosis
Skewness	Skewness	Skewness	Skewness	Skewness	Skewness
Rate	Rate	Rate	Rate	Rate	Rate

3.2. Wear Data preprocessing

All the steps involved in the wear data preprocessing are shown in Fig. 9.

The original, denoised, pre-machining tool image (Fig. 9(a)) was captured along with an intermediate, denoised tool image. The colour images were then converted into binary format (pixels values either 0 or 255) which are shown in Fig. 9(b). In order to remove the

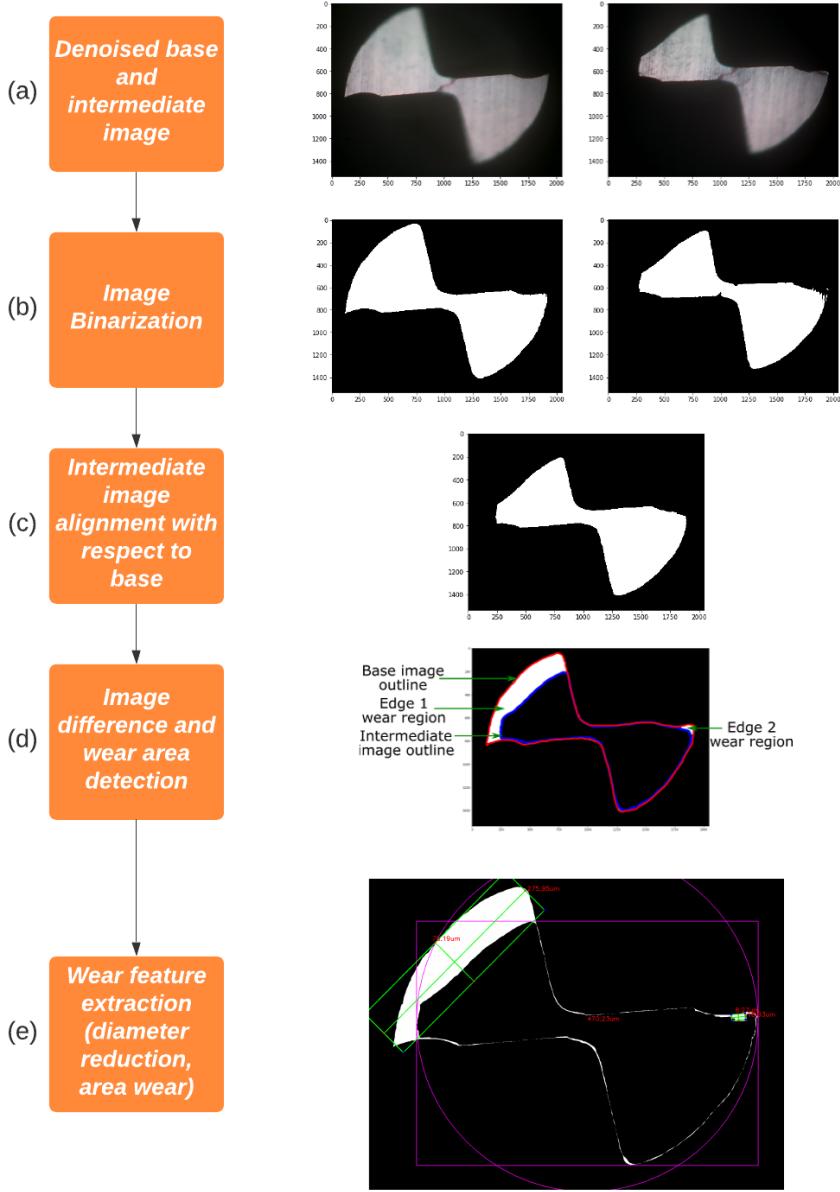


Figure 9: Wear data processing flow (a-e, steps involved)

unwanted features, the binarized images were subjected to a few iterations of dilation-erosion. The dilation procedure adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries. Fig. 9(c) shows an intermediate image aligned with respect to the base image. This was done to obtain an accurate measurement of the wear regions. Finally, Fig. 9(d) shows the image obtained after taking a difference between the base and the intermediate image. The representative tool image in Fig. 9(d) shows the boundaries of the base and intermediate images along with

the detected wear region on each cutting edge. This image was used to calculate the contours and subsequently the bounding boxes as shown in Fig. 9(e). The measurements of the extracted wear features were then made in pixels and subsequently converted to micrometer units using a parameter called pixels-per-metric (ppm). It was defined using the tool diameter value in micrometers ($500 \mu\text{m}$) and the tool diameter measured in pixels from the base image.

A sample dataset of the extracted wear features is shown in Table. 5. $orE1$ and $orE2$ represent the orientation of the measured wear (1 is chipping and 2 is flank wear) for each cutting edge. $wear1Ei$ (in μm) represents the length, whereas $wear2Ei$ (in μm), the breadth, of the rectangular wear area bounding box for each cutting edge i (equal to 1 for left edge and 2 for right edge). $wearAEi$ (in μm^2) is wear of chipped area for each edge.

Table 5: Snapshot of wear data

slot	$orE1$	$orE2$	$dRed$	$wearAE1$	$wear1E1$	$wear2E1$	$wearAE2$	$wear2E1$	$wear2E2$
5	1	1	34.7	330.25	61.9	8.53	2169.27	174.58	54.14
10	1	1	39	424.61	66.18	10.71	2148.64	167.53	26.19
15	1	1	67.31	676.64	73.94	16.42	9849.77	266.93	62
20	1	1	66.86	433.2	50.67	13.94	9760.28	266.67	61.27
25	1	1	68.92	574.36	70.37	15.83	9920.73	266.88	62.29

3.3. Computing the Static Radial Tool Runout

The pre-machining images captured after installing on the machining center, at a rotation interval of 30° , were used to compute the static radial runout of the tool. Fig. 10(a) shows the binarized image obtained from the colour image which was used for edge detection (Fig. 10(b)). The detected edges were then used for obtaining the contours and the bounding box (Fig. 10(c)). The images thus obtained were grouped in pairs with a phase difference of 180° . The distance between centers of such pairs, divided by two, gives the radial runout value (Fig. 10(d)).

3.4. Computing wear classes and remaining-useful-life

After obtaining the diameter reduction and area wear values for all the intermediate tool images, the tool state classification was done as described by Eq. 2 for diameter

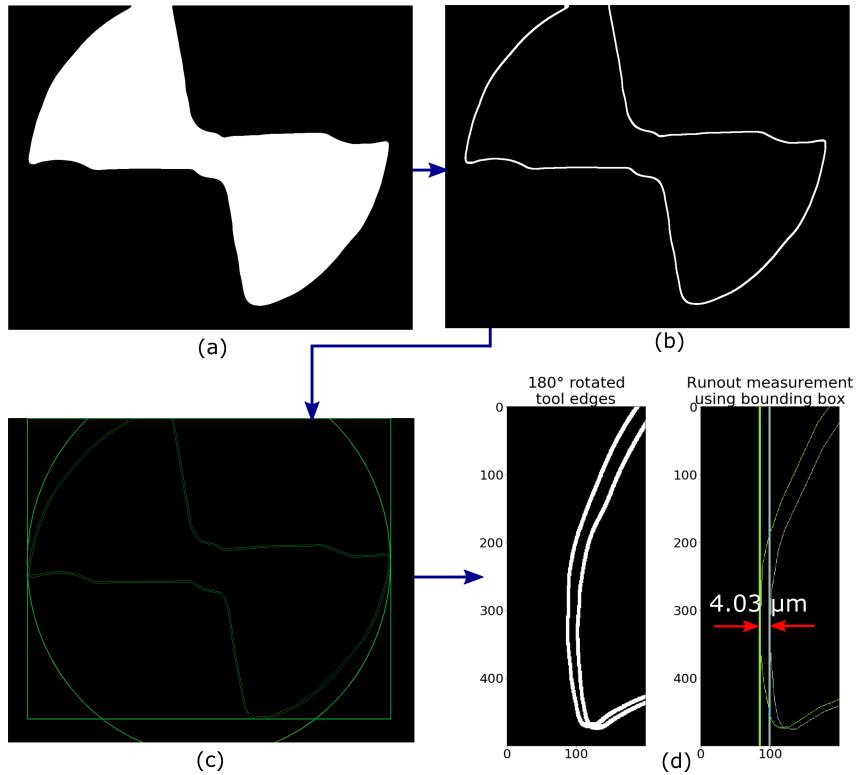


Figure 10: The process flow to compute static radial tool runout: (a) Binarized tool image (b) Edge detection of the tool (c) Contour and bounding box detection (d) Runout calculation

reduction and by using Eq. 3 for edge area wear. Class 1 defines the rapid initial wear region, Class 2 defines the steady-state wear and Class 3 defines the critical wear to failure region. The remaining-useful-life (RUL) for a particular data-point was computed by identifying the first time instant when the tool in observation reached the end-of-tool-life criteria and then taking a difference with the time instant of that data point.

$$Diameter\ Reduction \underset{classification\ criteria}{=} \begin{cases} Class\ 1 & \text{if diameter reduction} \leq 60\mu m \\ Class\ 2 & \text{if } 60\mu m < \text{diameter reduction} \leq 125\mu m \\ Class\ 3 & \text{if diameter reduction} > 125\mu m \end{cases} \quad (2)$$

$$Edge\ Wear\ Area \underset{classification\ criteria}{=} \begin{cases} Class\ 1 & \text{if wear area} \leq 5000\mu m^2 \\ Class\ 2 & \text{if } 5000\mu m^2 < \text{wear area} \leq 15000\mu m^2 \\ Class\ 3 & \text{if wear area} > 15000\mu m^2 \end{cases} \quad (3)$$

The wear state classification and RUL was added to the preprocessed dataset to be used as output labels for model training. At this point, the process parameters pertaining to each data-point in the dataset were also added as input features for the model.

4. Model Architectures and Training

The workflow for the model training process is shown in Fig. 11. The two neural-network architectures employed for training are discussed in sec. 4.1 and sec. 4.2.

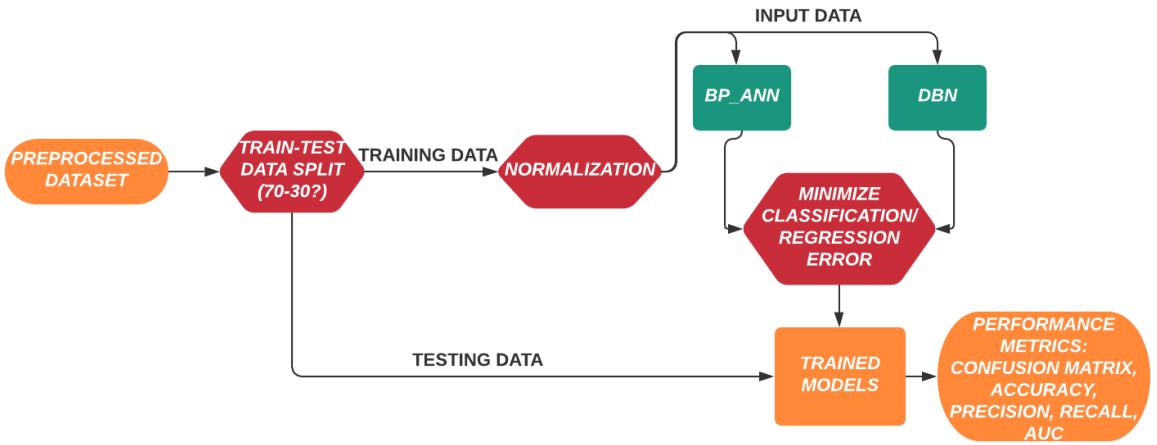


Figure 11: The model training process flow

The dataset obtained after preprocessing was split into train-validation-test sets with the train-test split being 7:3 and train-validation split being 4:1. Separate train-test split was done for each output label: (i) Diameter reduction, (ii) Area wear of edge 1, (iii) Area wear of edge 2 and (iv) Remaining useful life (RUL); and thus four different pairs of training and testing datasets are obtained. The dataset was then normalized to range [0,1] using min-max-scaler.

The input data consisted of 63 features, including the process parameters and the extracted features from the sensor data. The output in each case was dependent on the prediction criteria. For wear state classification based on diameter reduction/area wear of edges, the output labels were one-hot encoded (converted from decimal to binary format) before feeding in for training. For RUL prediction, the output labels were the time to end-of-tool-life (tteotl) value calculated for each datapoint. Subsequently, the artificial

neural networks were trained for wear state classification as well as RUL prediction and deep-belief-network was trained for diameter reduction based wear state classification. The architectures implemented for both the models are described below.

4.1. Neural Network

The architecture used for training the backpropogated neural-network model is displayed in Fig. 12.

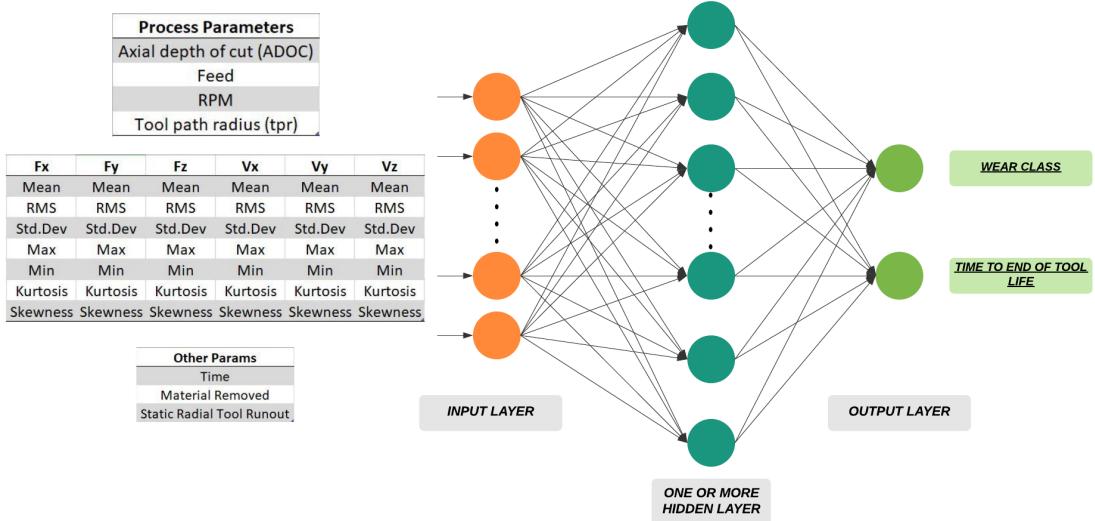


Figure 12: Architecture for neural network

A single dense (fully-connected) hidden layer, with 63 neurons formed the first layer in the architecture. In the case of regression for RUL prediction, another hidden layer with 200 neurons was added after the first layer. A *relu* activation function was used at the output of the hidden layers. The *relu* function is a piecewise linear function that will output the input directly if is positive and zero otherwise. To prevent overfitting of the training data, a regularization layer with dropout parameter set to 0.2 (20% of the data would be dropped at random) was added after the hidden layer. Finally, the output layer consisted of 3 neurons, one for each wear state during classification. A softmax activation function was used to convert the logits (a vector of raw, non-normalized predictions that a classification model generates) for each of the wear state into probability values. During regression, the output label was the RUL and was computed directly without the need of any additional activation functions after the output layer.

The training was done in mini-batches of size 32 and a total of 100 epochs were used for weight optimization. An epoch is completed when the entire training set passes through the network and contributes to the weight updates. One of the key features of this model training was the usage of Adam optimizer to minimize the classification/regression prediction error. Adam optimization is a stochastic gradient descent (SGD) method that is based on adaptive estimation of first-order and second-order moments. It adapts the learning rate of each weight individually. The moment estimation for the current batch (t) is done as in Eq. 4. Stochastic gradient descent is a technique which randomly picks one data point from the entire dataset to compute the error gradients in each iteration, with the goal of significantly reducing the large computation overhead of the standard gradient descent technique.

$$First\ moment : \begin{cases} Biased & m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \\ Unbiased & \hat{m}_t = \frac{m_t}{1 - \beta_1^t} \end{cases} \quad (4a)$$

$$Second\ moment : \begin{cases} Biased & v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \\ Unbiased & \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \end{cases} \quad (4b)$$

Here, $g(t)$ is the gradient of the current batch and β_1, β_2 (and also ϵ) are the control parameters for this optimizer. The values of $\beta_1 = 0.9$ and $\beta_2 = 0.999$ ($\epsilon = 1e - 7$) were used. The weight update is then made using the Eq. 5

$$w_t = w_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \quad (5)$$

Here, η is the step size. The trained model is saved alongwith the weights and biases and finally evaluated on the testing set.

4.2. Deep Belief Network

The DBN architecture employed for this study is shown in Fig. 13. The individual components of the architecture and the training process are discussed below.

Deep Belief Networks employ a hierachial structure with multiple stacked restricted boltzmann machines (RBMs) (Fig. 14(a)) and use a layer-by-layer learning technique followed by a fine-tuning procedure. Every layer in a deep belief network (Fig. 14(b)),

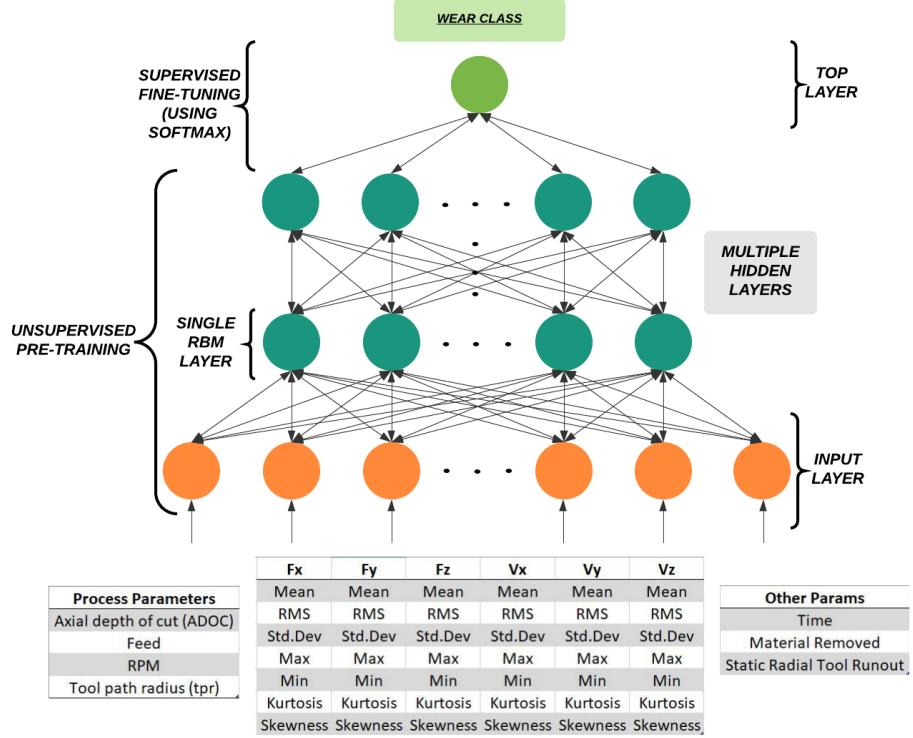


Figure 13: Architecture for deep-belief network

except the input and output, has a double function. It serves as a hidden layer to the previous layer and as a visible layer to the next layer. The training process consists of a pre-training, fine-tuning and prediction.

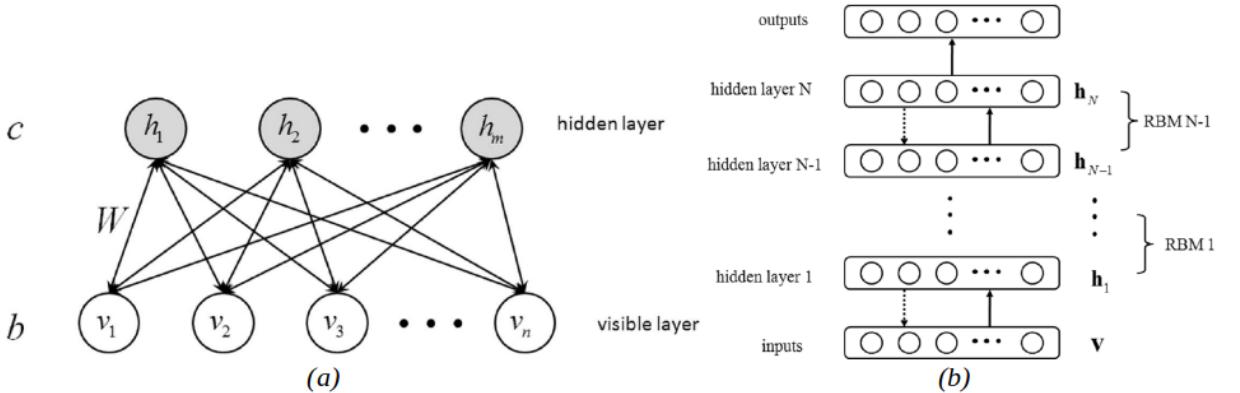


Figure 14: The structure of deep belief networks (a) restricted boltzmann machine (RBM) (b) stack of multiple RBMs to form a DBN [9]

RBM (Fig. 14(a)) are a type of generative stochastic artificial neural network and are effective in feature extraction for initializing the feed-forward neural network. There

are two layers in an RBM namely the visible and the hidden layers and each neuron in these layers has a binary value (0 or 1). Each neuron in the visible layer is connected to the neurons in the hidden layer in an undirected way with no connections within a layer. Each connection has a weight associated with it and is denoted by $\mathbf{w} = w_{ij}$ such that it is the undirected weight associated with hidden neuron h_j and visible neuron v_i . The biases associated with the visible and hidden neurons are denoted by vectors \mathbf{b} and \mathbf{c} , respectively. The set of parameters including the weights and biases is denoted by θ [9]. The probability of neuron having value 0 or 1 is calculated based on the energy Eq. 6. The energy of the joint configuration is given by Eq. 7, where I and J are the number of visible and hidden units.

$$\text{For hidden neuron } h_j : E_j = \sum_{i=1}^n w_{ij} v_i + c_j \quad (6a)$$

$$\text{For visible neuron } v_i : E_i = \sum_{j=1}^m w_{ij} h_j + b_i \quad (6b)$$

$$E(\mathbf{v}, \mathbf{h}; \theta) = - \sum_{i=1}^I \sum_{j=1}^J w_{ij} v_i h_j - \sum_{i=1}^I b_i v_i - \sum_{j=1}^J c_j h_j \quad (7)$$

The model assigns a probability value to a visible vector (\mathbf{v}) as given by Eq. 8. The final values of the neurons in the visible vector are determined by choosing a random number (u) to define a step function for h_j , where it is equal to 1 if $P(h_j = 1) \geq u$, and 0 otherwise.

$$P(\mathbf{v}; \theta) = \sum_{\mathbf{h}} P(\mathbf{v}, \mathbf{h}; \theta) = \frac{1}{\sum_{\mathbf{h}} \sum_{\mathbf{v}} e^{-E(\mathbf{v}, \mathbf{h}; \theta)}} \sum_{\mathbf{h}} (e^{\mathbf{v}^T \mathbf{w} \mathbf{h} + \mathbf{b}^T \mathbf{v} + \mathbf{c}^T \mathbf{h}}) \quad (8)$$

The pre-training involves using the stochastic gradient descent (SGD) technique to optimize the weights and the biases by calculating the negative log probability (Eq. 9 [9]). Each hidden layer is trained using the activation probabilities of the hidden layer of the lower RBM as input data, and its own output serves as the input for the upper RBM. For this unsupervised, layer-by-layer training, a total of 10 epochs demonstrated optimal performance.

$$\left\{ \begin{array}{l} \frac{\partial \ln P(\mathbf{v};\theta)}{\partial w_{ij}} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model} \\ \\ \frac{\partial \ln P(\mathbf{v};\theta)}{\partial \mathbf{c}} = \langle h_j \rangle_{data} - \langle h_j \rangle_{model} \\ \\ \frac{\partial \ln P(\mathbf{v};\theta)}{\partial \mathbf{b}} = \langle v_i \rangle_{data} - \langle h_i \rangle_{model} \end{array} \right. \quad (9)$$

A softmax layer was added after the final hidden layer to convert the raw output values to probabilities of wear classes. The fine-tuning of model weights and biases involves using backpropagation which forms the supervised learning step to minimize the classification error.

5. Results and Discussions

This section is divided into two parts: Section 5.1 discusses the results from the analysis done based on the experimental procedure. Section 5.2 details the model performances and the characteristics of the prediction results.

5.1. Experimental results

The tool wear progression and force analysis is discussed from an overall point of view as well as in terms of variation with tool path radius. The tool ids (Table 3): T80, T82, T83, T85 and T86 broke prematurely while T79, T81 and T84 reached the end-of-life criteria. Tool ids T82, T85 and T86 suffered catastrophic failure after only a few seconds of machining.

5.1.1. Variation of static runout and peak-to-peak forces

The analysis of the pre-machining tool images, after installing in the machining center, yielded the static radial tool runout. The runout values measured at different angles and the mean runout value for each experiment are described in Table 6. The calculated values of static radial tool runout for T83 ($7.03 \mu m$) and T86 ($5.57 \mu m$) were considerably large. This could be a potential reason for their catastrophic failure.

Table 6: Static radial tool runout

Tool No	val_1 (μm)	val_2 (μm)	val_3 (μm)	val_4 (μm)	val_5 (μm)	val_6 (μm)	Average (μm)
T_{79}	1.95	4.03	2.21	—	—	—	2.73
T_{80}	6.29	6.82	6.27	8.39	8.24	8.67	7.45
T_{81}	2.13	2.41	2.5	2.17	2.77	—	2.39
T_{82}	5.2	5.12	5.3	5.43	5.79	—	5.37
T_{83}	7.81	7.2	7.52	4.81	7.09	7.78	7.03
T_{84}	3.38	3.28	4.88	4.36	4.31	4.67	4.15
T_{85}	6.3	5.58	5.5	6.07	5.44	6.01	5.82
T_{86}	5.39	5.75	5.24	5.44	5.77	5.87	5.57

The peak-to-peak force was analyzed for the entire duration of machining for all the experiments as shown in Fig. 15. A sudden drop in force values was observed around 700 seconds of machining for T79 (Fig. 15(a)) indicating that the tool entered a secondary steady state region before reaching end-of-life. Forces for T81 (Fig. 15(c)) showed a higher magnitude in the x-direction whereas for T80, T83 and T84 (Fig. 15(b,e,f)) the magnitude in the y-direction was higher than the other directions. Fig. 15(d,g,h) shows a similar rate of increase in force magnitude in all the three directions before the catastrophic failure of the tools.

5.1.2. Variation of tool wear with tool path radius

Three different metrics were used to quantify tool state: (i) Diameter reduction; (ii) Chipped area of edge 1; (iii) Chipped area of edge 2. The individual edge-wear quantification helps in better understanding the tool state. The wear progression curves, in terms of diameter reduction with distance for different tpr is shown in Fig. 16. For tpr of 250 μm (Fig. 16(a)), the tools entered the steady state region after machining a distance of 100-150 mm. However for tpr of 500 μm (Fig. 16(b)) the tools (T83 and T84) were able machine a distance of 60-100 mm before entering steady state region (T84) or breaking (T83). On the other hand, T82 entered the critical wear region after machining only 10-15 mm and then broke. The diameter reduction (Fig. 16(c)), after having machined only 5-10 mm, was even more severe for tpr of 750 μm . This indicates that the rate of tool

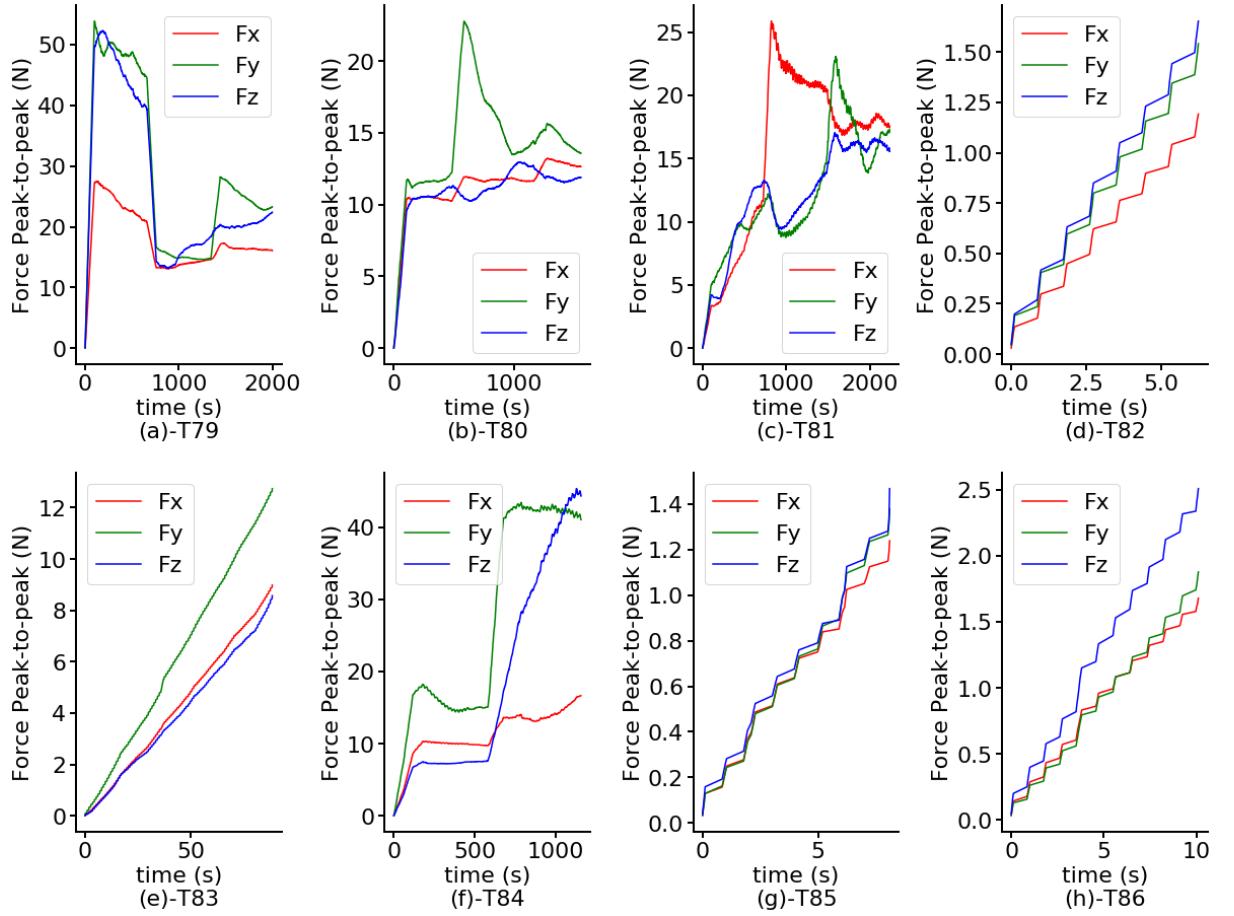


Figure 15: Peak-to-peak force variations: ([a-h] for individual experiments)

wear progressively increases with increase in tool path radius.

The tool condition for each of the three tool path radii at different wear stages (I - Fresh tool, II - Rapid initial wear, III - Steady state wear and IV - Before breakage) is shown in Fig. 17. The variation in tool condition with $250 \mu\text{m}$ tpr showed rounding of cutting edges followed by severe flank wear (Fig. 17(a-II,III)) before breakage. Significant adhesive wear (Fig. 17(b-II,IV)) was observed for $500 \mu\text{m}$ tpr before breakage. For $750 \mu\text{m}$ tpr, excessive chipping of edges (Fig. 17(c-II)) along with some adhesion was observed.

The analysis of individual edge wear for different tpr (Fig. 18) shows that, at any given instant, there is only one cutting edge in contact with the workpiece. For both $250 \mu\text{m}$ tpr (Fig. 18(a)) and $500 \mu\text{m}$ tpr (Fig. 18(b)), the first shift of contact from edge 1 to edge 2 was observed within 70-100 seconds of machining. The subsequent shifts varied from experiment to experiment. For $750 \mu\text{m}$ tpr (Fig. 18(c)) no switch was detected as

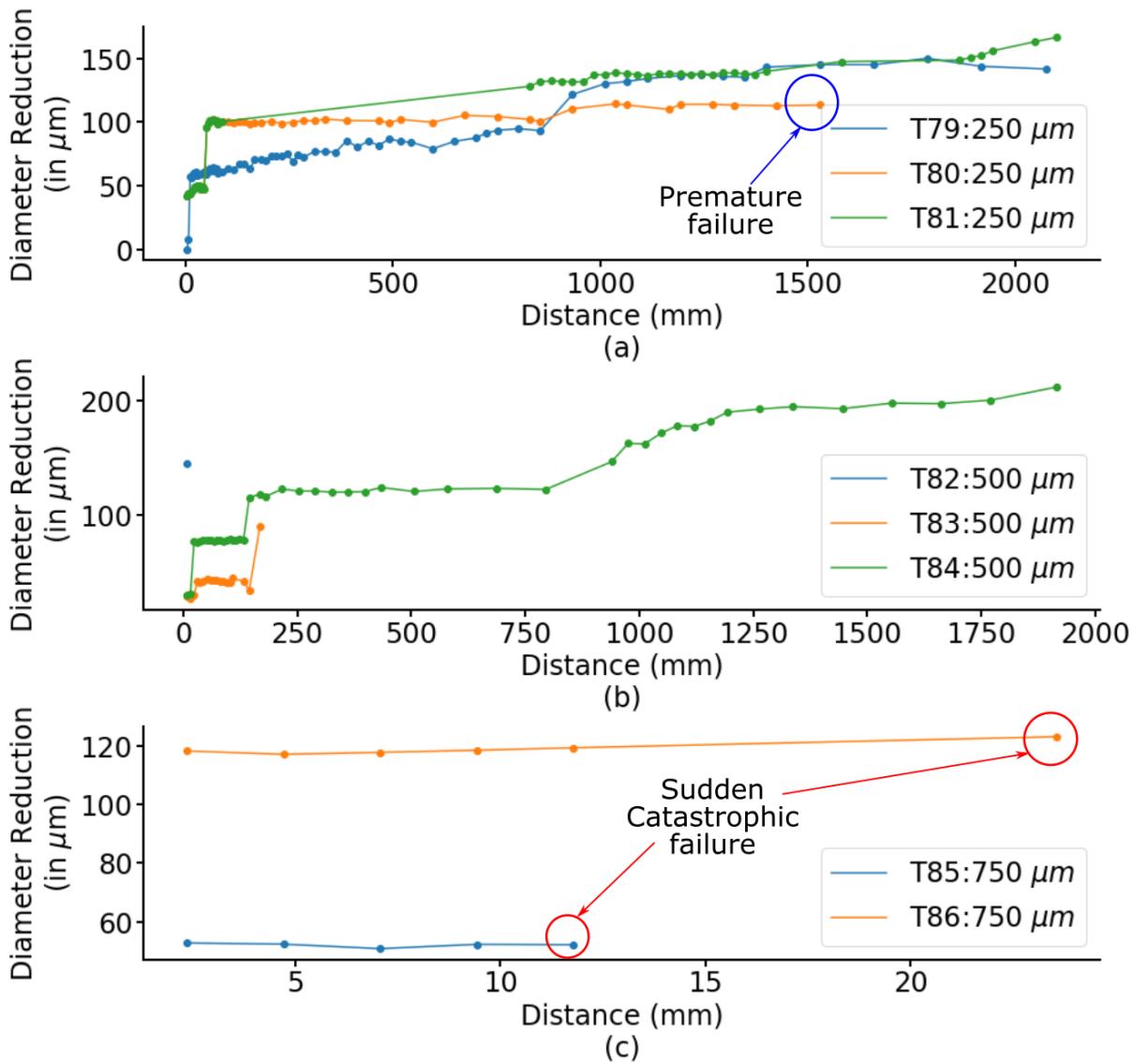


Figure 16: Variation in diameter reduction with distance: (a) 250 μm tpr (b) 500 μm tpr (c) 750 μm tpr

the tools suffered a catastrophic failure after machining for only 10-15 seconds.

5.2. Model performances and predictions

Two predictive models were used for training including the ANN and DBN. The training set consisted of 18737 data points while the testing set consisted of 10038 data points for both the models. The performances of both ANN and DBN are evaluated and discussed in the following sections.

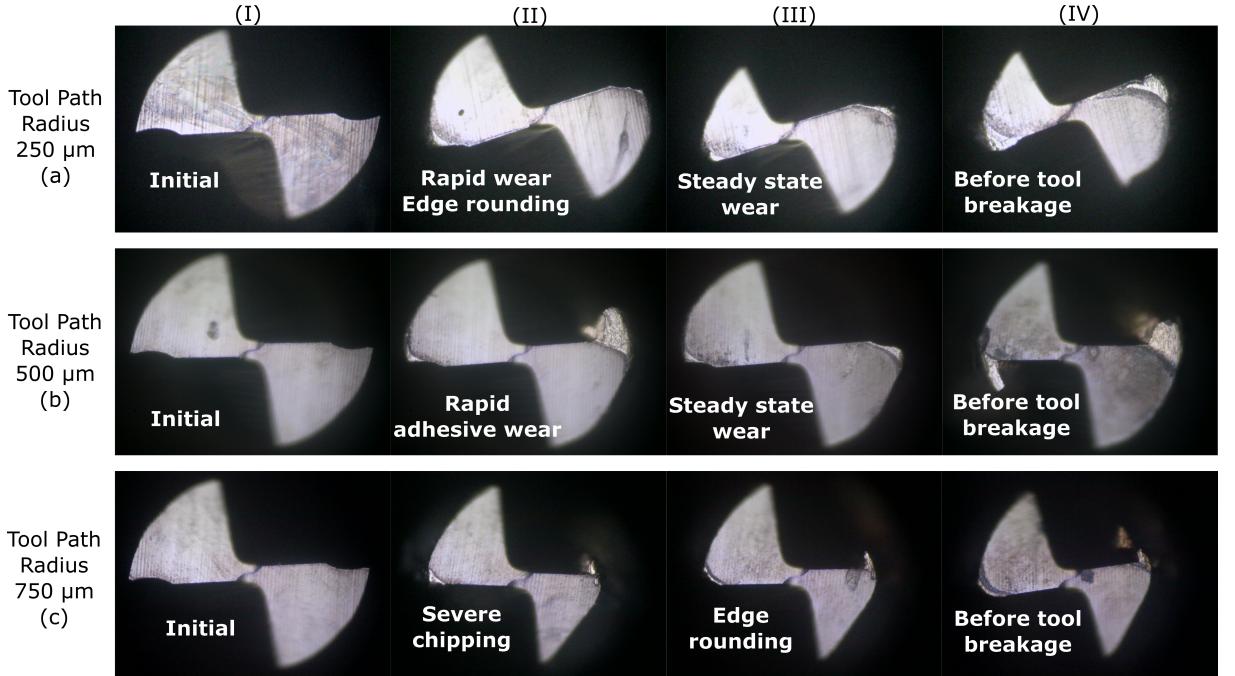


Figure 17: Tool images at different stages for different tpr: (a) $250 \mu\text{m}$ tpr (b) $500 \mu\text{m}$ tpr (c) $750 \mu\text{m}$ tpr

5.2.1. ANN performance

The ANN architecture involved 4 different output parameters. The diameter reduction and chipping area of each cutting edge were used for wear state classification. Fig. 19(a) shows the evolution of various performance metrics as the model is trained. Fig. 19(a-I) shows the variation of the categorical cross-entropy loss with training epochs. Categorical cross-entropy loss function is a measure of how distinguishable two discrete probability distributions are from each other. Higher value of loss indicates that the classification model is not able to accurately distinguish between different classes. The AUC curve (Fig. 19(a-II)) is another measure of separability where a higher value indicates that the model is able to accurately distinguish between classes. Precision (Fig. 19(a-III)) is defined as the fraction of predictions for a particular class that were actually correct. Finally, recall (Fig. 19(a-IV)) is defined as the fraction of the total number of data points belonging to a particular class that were accurately predicted. The remaining-useful-life (RUL) was predicted using a regression model and the statistics used include mean squared error (b-I,II), mean absolute error (b-III) and the cosine-value (b-IV) (Fig. 19(b)). A larger

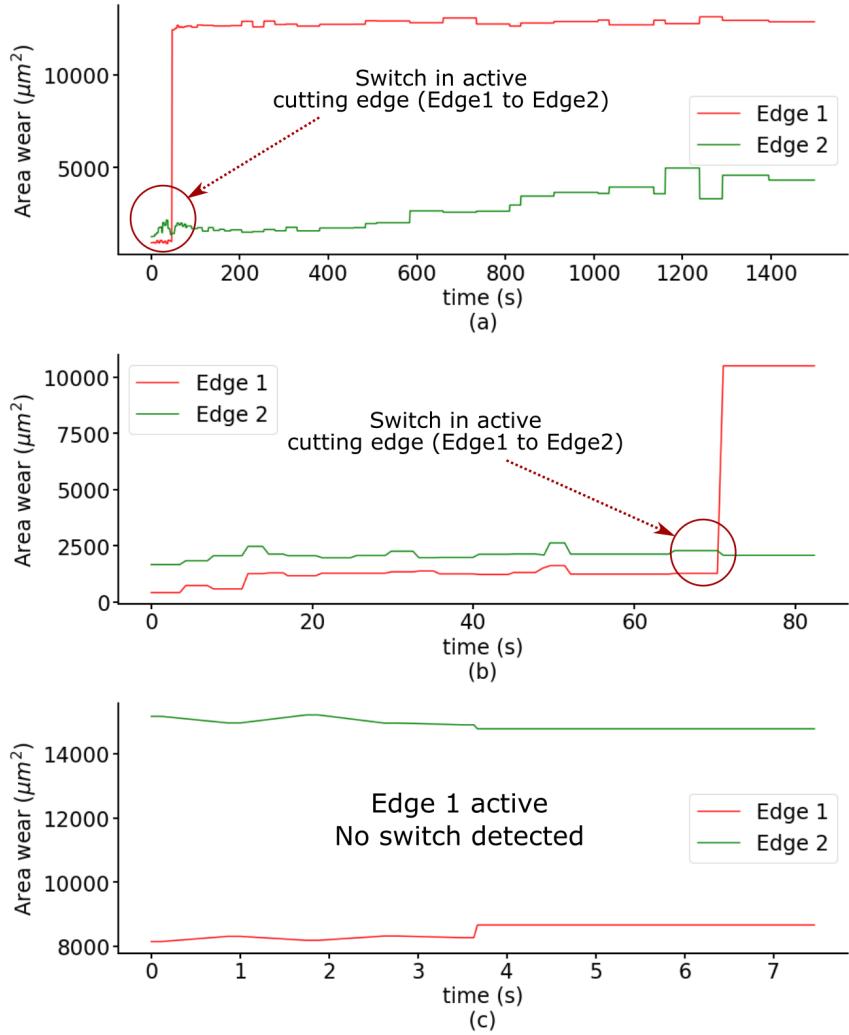


Figure 18: Variation of area wear of individual cutting edges with time for different tpr: (a) $250 \mu\text{m}$ tpr (b) $500 \mu\text{m}$ tpr (c) $750 \mu\text{m}$ tpr

cosine-value indicates a higher correlation between predicted and actual values.

The model was then evaluated on unseen testing data and the results obtained are detailed in Table. 7. It is observed that the ANN is able to predict wear state related to each cutting edge with a slightly better accuracy than the state related to diameter reduction. This can be expected as the diameter reduction metric is a function of both the cutting edge wear values and therefore would have a compounded error in prediction than the individual edge wear.

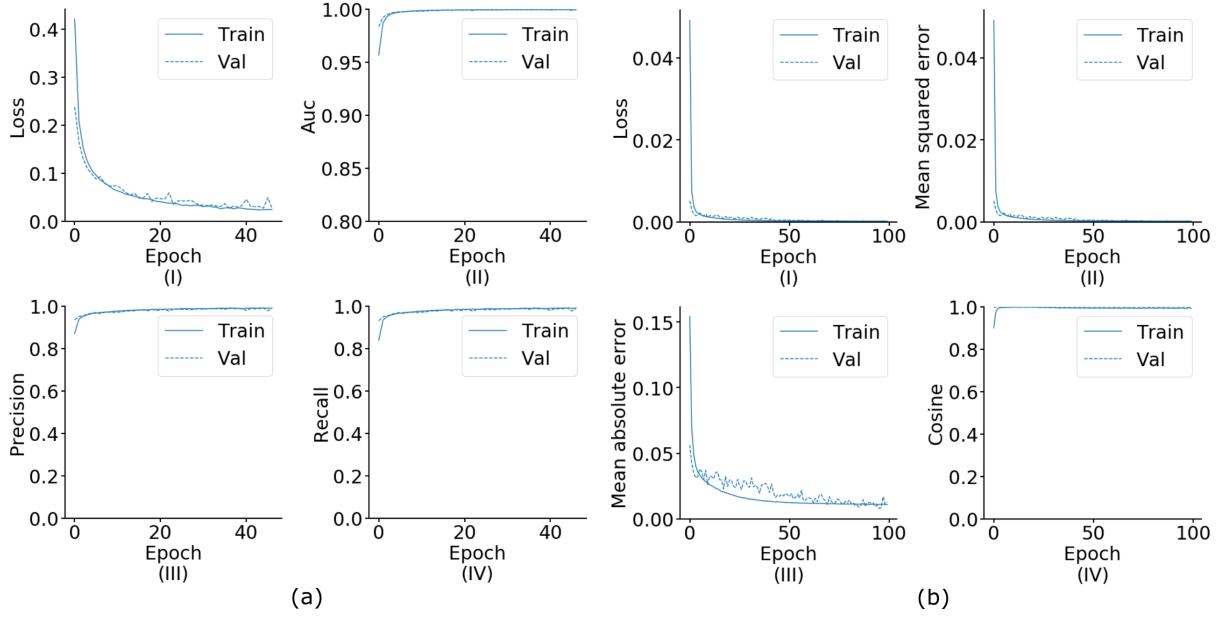


Figure 19: Training statistics for ANN (a) Diameter reduction (b) RUL

Table 7: ANN Model performance metric on testing dataset

Output Label	Accuracy	Precision	Recall	MSE	MAE	Cosine
Diameter Reduction	0.9873	0.9812	0.9808	—	—	—
Area Wear E1	0.9945	0.9918	0.9917	—	—	—
Area Wear E2	0.9923	0.9885	0.9884	—	—	—
RUL	—	—	—	11.38	0.6451	0.9634

The confusion matrices (displaying variation of true label with predicted label) pertaining to the three output labels are shown in Fig. 20. For the diameter reduction output label (Fig. 20(a)), 263 out of 304 labels in the initial wear region, 5392 out of 5426 labels in the steady state region and 4302 out of 4308 labels in the critical wear region were accurately classified. For the edge 1 area wear output label (Fig. 20(a)), 1648 out of 1667 labels in the initial wear region, 6949 out of 6973 labels in the steady state region and 1292 out of 1398 labels in the critical wear region were accurately classified. For the edge 2 area wear output label (Fig. 20(a)), 3797 out of 3813 labels in the initial wear region, all the labels in the steady state region and 1482 out of 1487 labels in the critical wear region were accurately classified.

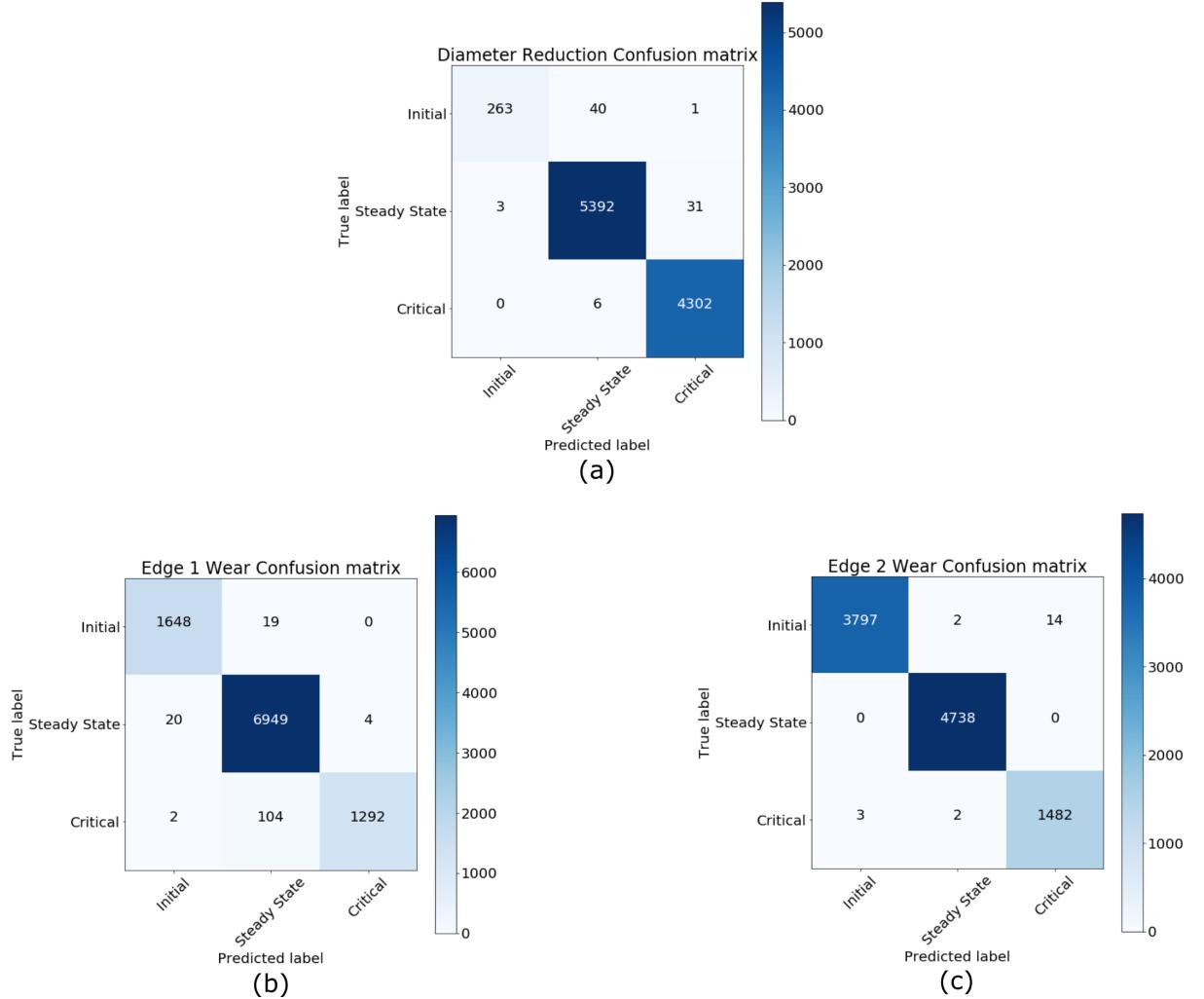


Figure 20: ANN Confusion matrices (statistics of true vs predicted labels)

The trained regression model was tested on each experiment's dataset and the resulting RUL values were predicted. Fig. 21 shows the true and predicted RUL values for all the experiments. For T79, T80 and T84 (Fig. 21(a,b,e)), the predicted RUL values at any time instant were underestimated than the actual values. However, for T81, T83 and T86 (Fig. 21(c,d,f)), overestimation in prediction was observed during certain time intervals.

5.2.2. DBN performance

The training of the DBN required a more intensive hyperparameter tuning. The hyperparameters that were taken into account for tuning included the learning rate and number of iterations for the unsupervised pre-training, the learning rate and number of iterations backpropagation, the number of hidden layers and number of neurons per layer.

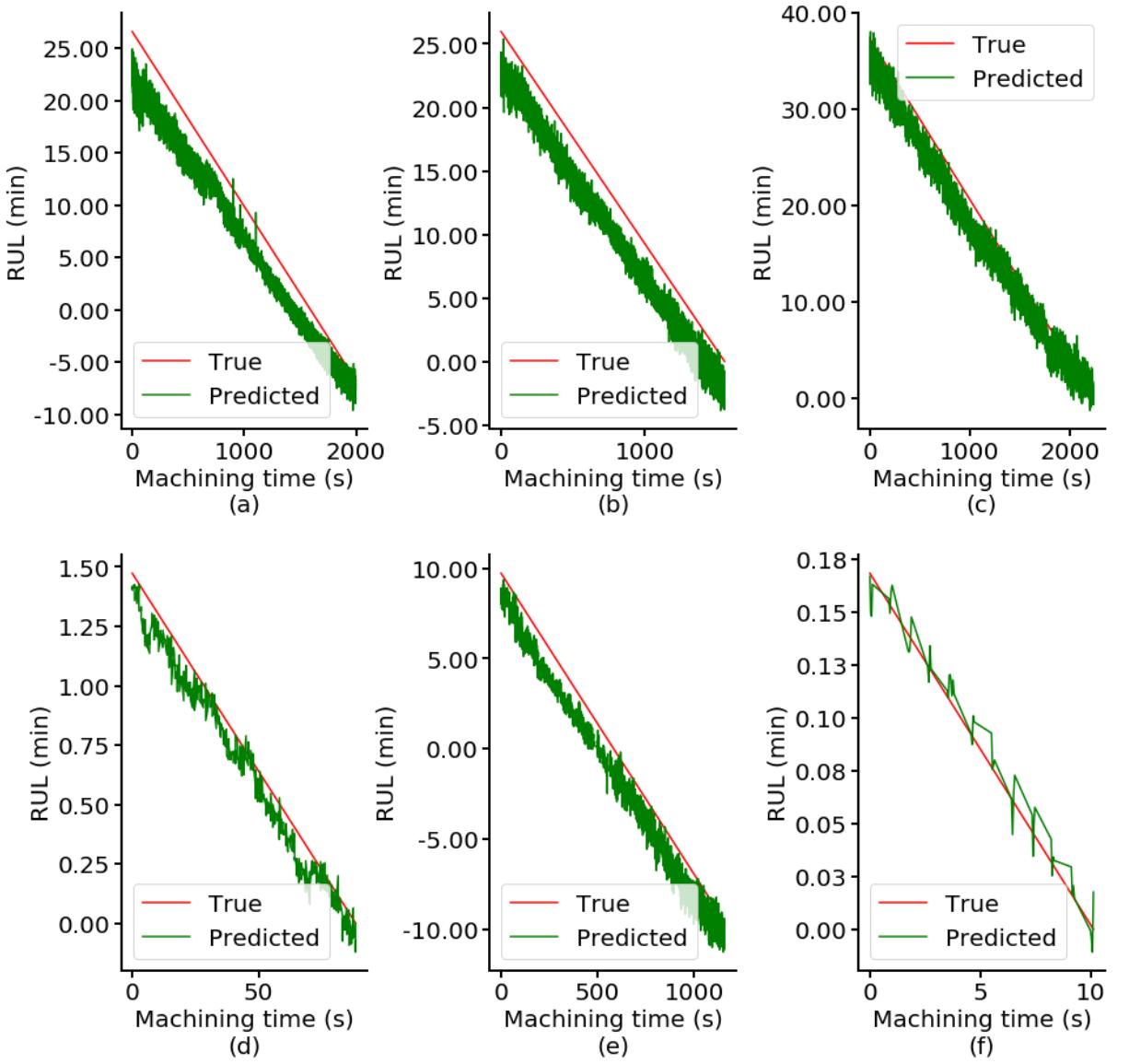


Figure 21: ANN RUL predictions

After training 270 models using different combinations of the hyperparameters mentioned above and cross-validation (RSCV and GSCV), decent performance was obtained for classification with diameter reduction as output.

The model performance under four different hyperparameter combinations is shown in Fig. 22. For the training statistics shown in Fig. 22(a,b), the hidden layers structure was kept the same with 2 hidden layers and 63 neurons in each layer along with the other parameters. The backpropagation learning rate of 0.01 (Fig. 22(b)) displayed a *hockey-stick effect* indicating that the learning rate was too small. The *hockey-stick effect* is

characterized by a sharp rise or fall of data points after a long flat period and occurs due to poor initialization of the parameters. This effect was not observed when the learning rate was 0.2 (Fig. 22(a)), however, the variability in classification error was too high which indicated a very large learning rate. By keeping the learning rate at 0.05 but increasing the number of hidden layers to 3 (Fig. 22(c)), some improvement was observed in terms of a decrease in the length of the *hockey-stick effect* and the variability. A further improvement was observed when 2 hidden layers were used with 200 neurons in the second hidden layer (Fig. 22(c)). The final structure of the DBN that was used for training is described in Table. 8 and the classification report from testing data is described in Table. 9.

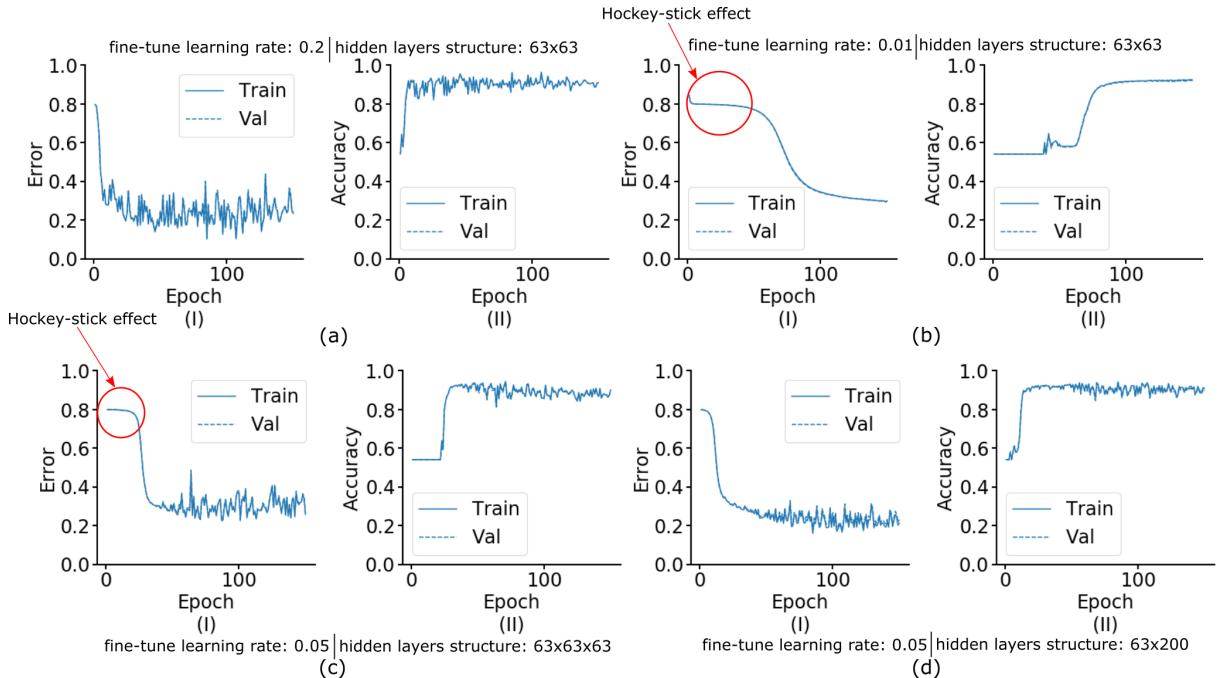


Figure 22: DBN hyperparameter tuning (loss and accuracy curves)

Table 8: DBN final architecture details

Hidden Layer Structure	RBM learning rate	Fine-tuning learning rate	RBM epochs	Fine-tuning epochs	Batch Size	Activation function
63x200	0.01	0.05	10	150	400	relu

Table 9: DBN testing statistics for diameter reduction based classification

Class	Precision	Recall	F1-score	Support
Initial Wear	0.87	0.26	0.4	304
Steady State Wear	0.91	0.95	0.93	5426
Critical Wear	0.93	0.93	0.93	4308
Accuracy	—	—	0.92	10038
Macro Average	0.9	0.71	0.75	10038
Weighted Average	0.92	0.92	0.91	10038

The relatively lower accuracy of DBN(93%), compared to the ANN(98%), indicates the possibility that the set of hyperparameters used may not be ideal. As can be seen from Fig. 22(d), some amount of variability is still present in the model with the chosen set of hyperparameters (Table. 8). In order to obtain a smooth convergence of error with no *hockey-stick effect*, a significantly larger number of different hyperparameter combinations need to be used to train and evaluate an equal number of models. This process demands higher computational power to achieve reasonable training times which was unavailable during this research. Also, the high value of precision but a drastically lower recall for the initial wear region indicates that the model is able to distinguish this state from the other states correctly but is unable to accurately predict it (find the boundary between initial wear and steady state wear). This is primarily because of the relatively lower support-value for this class and suggests that tool wear sampling in the initial phases (first 200 slots) of the experiment be done at a higher frequency to obtain the exact differential in tool condition.

6. Conclusions

This paper investigated the impact of tool path radius on cutting tool wear in micro end-milling processes. A unique set of experiments involving semi-circular tool paths, varying spindle speed and feed was designed. The process of generating g-codes for the machining the slots in different batch sizes was automated. Cutting data was generated from sensors in the form of force, acceleration and vibration velocity. The static radial tool runout and intermediate images of the tool face were captured to generate tool wear

data. Sensor data was segmented to isolate individual slot data using k-means clustering and peaks-analysis algorithms. Subsequently, various statistical features were extracted by grouping the sensor data of each slot into 4 segments which generated a total of 33460 data points.

The denoised, coloured tool images were binarized and aligned with respect to base image. The final image for measuring the wear features was obtained by taking a difference between the binarized intermediate tool image and the binarized base image. The extracted features from the final image included the chipped area of individual cutting edges and diameter reduction values. Tool state was classified into three regions based on diameter reduction and chipped area of individual cutting edges by defining specific criteria for each. The remaining-useful-life for a particular data point was calculated by taking the difference between the first time instant when the tool reached the end-of-tool-life criteria and the time instant of that data point.

The progression of tool wear was modelled using artificial neural network (ANN) and deep belief network (DBN) to capture the non-linear relation of tool path radius along with the other process parameters and sensor data. For classification with ANN, single hidden layer with 63 neurons were used while two hidden layers with 63 and 200 neurons each were used for DBN. Regression analysis for predicting RUL was done using ANN with two hidden layers with 63 and 200 neurons each. Adam optimizer was used for training ANN to improve rate of convergence of prediction error. The DBN training involved an unsupervised pre-training using 10 iterations for initializing the weights and the biases. This was followed by a supervised fine-tuning of the weights and biases using stochastic gradient descent technique.

The training performance of the model was improved using hyperparameter tuning techniques. Hyperparameters that were tuned included the learning rate, number of training iterations and hidden layers structure. This involved training models with 270 different hyperparameter combinations and the one with the best performance during training was selected. The final, trained models where then evaluated on the testing dataset to evaluate their performance. The following were the main conclusions drawn from this study.

- In the case where the tool broke, a significant rise in peak-to-peak, z-direction forces was observed. Similar force signatures were observed when tool reached end-of-life. Such force signatures can be used as indicators of nearing the end-of-tool-life or potential breakage.
- Wear rates for higher tool path radius ($500 \mu m$) were significantly higher than lower tool path radius ($250 \mu m$). For tpr of $750 \mu m$, sudden, catastrophic tool failure was observed after machining only 10-15 slots.
- Lower tpr showed edge rounding and significant flank wear of the tool before breakage. However for $500 \mu m$ tpr, adhesive wear was found to be the major wear mechanism along with edge rounding and flank wear. For higher tpr, severe chipping along with adhesive wear lead to catastrophic tool failure.
- The monitoring of individual edge wear revealed the switch in active cutting edges while machining. In some cases, there was a switch in active cutting edge just before tool breakage. Such a monitoring strategy could provide more reliable information about tool condition and thus enable better decision making during on-line machining.
- The wear state classification accuracy, for ANN model, with individual edge wear as output label was found to be higher (99%) than with diameter reduction as output label (98%). This indicates that monitoring the wear progression of individual cutting edges could provide better prognostics insights.
- Regression analysis for tool remaining-useful-life indicated both underestimation and overestimation in predictions with a cosine-value of 0.96. However, the degree by which the results were underestimated was much higher than the overestimated results indicating an overall conservative prediction nature.
- The testing of deep belief networks revealed a slight degradation in classification of wear state (93% accuracy) as compared to the backpropogated artificial neural networks (98% accuracy). This could be significantly improved by implementing a more rigourous and dense hyperparameter tuning technique.

References

- [1] A Banerjee and EV Bordatchev. Effect of circular tool path on cutting force profile in micro-end-milling. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 226(6):1589–1600, 2012.
- [2] Mohammad Malekian, Simon S. Park, and Martin B.G. Jun. Tool wear monitoring of micro-milling operations. *Journal of Materials Processing Technology*, 209(10):4903–4914, 2009.
- [3] X. Li, B. S. Lim, J. H. Zhou, S. Huang, S. J. Phua, K. C. Shaw, and M. Er. Fuzzy neural network modelling for tool wear estimation in dry milling operation. *Annual conference of the prognostics and health management society*, pages 1–11, 2009.
- [4] Sohyung Cho, Sultan Binsaeid, and Shihab Asfour. Design of multisensor fusion-based tool condition monitoring system in end milling. *The International Journal of Advanced Manufacturing Technology*, 46(5-8):681–694, 2010.
- [5] Zhenhua Wu. Cutting tool condition monitoring and prediction based on dynamic data driven approaches. In *International Manufacturing Science and Engineering Conference*, volume 56826, page V001T02A110. American Society of Mechanical Engineers, 2015.
- [6] Dazhong Wu, Connor Jennings, Janis Terpenny, Robert X Gao, and Soundar Kumara. A comparative study on machine learning algorithms for smart manufacturing: tool wear prediction using random forests. *Journal of Manufacturing Science and Engineering*, 139(7), 2017.
- [7] Cyril Drouillet, Jaydeep Karandikar, Chandra Nath, Anne-Claire Journeaux, Mohamed El Mansori, and Thomas Kurfess. Tool life predictions in milling using spindle power with the neural network technique. *Journal of Manufacturing Processes*, 22:161–168, 2016.
- [8] Chong Zhang, Geok Soon Hong, Huan Xu, Kay Chen Tan, Jun Hong Zhou, Hian Leng Chan, and Haizhou Li. A data-driven prognostics framework for tool remaining

useful life estimation in tool condition monitoring. In *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–8. IEEE, 2017.

- [9] Yuxuan Chen, Yi Jin, and Galantu Jiri. Predicting tool wear with multi-sensor data using deep belief networks. *The International Journal of Advanced Manufacturing Technology*, 99(5-8):1917–1926, 2018.
- [10] Zhixiong Li, Rui Liu, and Dazhong Wu. Data-driven smart manufacturing: tool wear monitoring with audio signals and machine learning. *Journal of Manufacturing Processes*, 48:66–76, 2019.
- [11] Jian-wei Ma, Guo-qing Hu, Zhen-yuan Jia, Ning Zhang, and Fu-ji Wang. Effect of geometric feature and cutting direction on variation of force and vibration in high-speed milling of tc4 curved surface. *The International Journal of Advanced Manufacturing Technology*, 95(5-8):2207–2218, 2018.
- [12] Rodrigo Henriques Lopes da Silva and Amauri Hassui. Cutting force and surface roughness depend on the tool path used in side milling: an experimental investigation. *The International Journal of Advanced Manufacturing Technology*, 96(1-4):1445–1455, 2018.
- [13] Kaining Shi, Ning Liu, Sibao Wang, and Junxue Ren. Effect of tool path on cutting force in end milling. *The International Journal of Advanced Manufacturing Technology*, 104(9-12):4289–4300, 2019.
- [14] Amin Dadgari, Dehong Huo, and David Swailes. Investigation on tool wear and tool life prediction in micro-milling of ti-6al-4v. *Nanotechnology and Precision Engineering*, 1(4):218–225, 2018.
- [15] Vipindas K and Jose Mathew. Wear behavior of tialn coated wc tool during micro end milling of ti-6al-4v and analysis of surface roughness. *Wear*, 424-425:165–182, 2019.

- [16] Yishun Wang, Bin Zou, and Chuanzhen Huang. Tool wear mechanisms and micro-channels quality in micro-machining of ti-6al-4v alloy using the ti(c7n3)-based cermet micro-mills. *Tribology International*, 134:60–76, 2019.
- [17] Ingo G. Reichenbach, Martin Bohley, Fabio J.P. Sousa, and Jan C. Aurich. Tool-life criteria and wear behavior of single-edge ultra-small micro end mills. *Precision Engineering*, 55:48–58, 2019.
- [18] Chakradhar Bandapalli, Bharatkumar Mohanbhai Sutaria, Dhananjay Vishnu Prasad Bhatt, and Kundan Kumar Singh. Tool wear analysis of micro end mills - uncoated and pvd coated tialn & altin in high speed micro milling of titanium alloy - ti-0.3mo-0.8ni. *Procedia CIRP*, 77:626–629, 2018.
- [19] L. Alhadeff, M. Marshall, and T. Slatter. The influence of tool coating on the length of the normal operating region (steady-state wear) for micro end mills. *Precision Engineering*, 60:306–319, 2019.
- [20] Abram Pleta, Durul Ulutan, and Laine Mears. An investigation of alternative path planning strategies for machining of nickel-based superalloys. *Procedia Manufacturing*, 1:556–566, 2015.
- [21] Dongsheng Liu, Ying Zhang, Ming Luo, and Dinghua Zhang. Investigation of tool wear and chip morphology in dry trochoidal milling of titanium alloy ti-6al-4v. *Materials (Basel, Switzerland)*, 12(12), 2019.
- [22] Samad Nadimi Bavil Oliaei and Yiğit Karpat. Influence of tool wear on machining forces and tool deflections during micro milling. *The International Journal of Advanced Manufacturing Technology*, 84(9-12):1963–1980, 2016.
- [23] Xiaohong Lu, Furui Wang, Zhenyuan Jia, Likun Si, Chi Zhang, and Steven Y. Liang. A modified analytical cutting force prediction model under the tool flank wear effect in micro-milling nickel-based superalloy. *The International Journal of Advanced Manufacturing Technology*, 91(9-12):3709–3716, 2017.
- [24] Ce Han, Ming Luo, Dinghua Zhang, and Baohai Wu. Mechanistic modelling of worn drill cutting forces with drill wear effect coefficients. *Procedia CIRP*, 82:2–7, 2019.

- [25] Daniel Frank Hesser and Bernd Markert. Tool wear monitoring of a retrofitted cnc milling machine using artificial neural networks. *Manufacturing Letters*, 19:1–4, 2019.
- [26] Aitor Duo, Rosa Basagoiti, Pedro J. Arrazola, Javier Aperribay, and Mikel Cuesta. The capacity of statistical features extracted from multiple signals to predict tool wear in the drilling process. *The International Journal of Advanced Manufacturing Technology*, 102(5-8):2133–2146, 2019.
- [27] Xian Wu, Liang Li, Ning He, Guolong Zhao, Feng Jiang, and Jianyun Shen. Study on the tool wear and its effect of pcd tool in micro milling of tungsten carbide. *International Journal of Refractory Metals and Hard Materials*, 77:61–67, 2018.
- [28] Abram Pleta, Farbod Akhavan Niaki, and Laine Mears. A comparative study on the cutting force coefficient identification between trochoidal and slot milling. *Procedia Manufacturing*, 26:570–579, 2018.
- [29] Omer Eker, Faith Camci, and Ian K. Jennions. Major challenges in prognostics: study on benchmarking prognostic datasets. 2012.
- [30] Lei Zhang, Xingshan Li, and Jinsong Yu. A review of fault prognostics in condition based maintenance. *Spie The International Society For Optical Engineering*, 2006.
- [31] Ying Peng, Ming Dong, and Ming Jian Zuo. Current status of machine prognostics in condition-based maintenance: a review. *The International Journal of Advanced Manufacturing Technology*, 50(1):297–313, 2010.
- [32] Rui Zhao, Ruqiang Yan, Zhenghua Chen, Kezhi Mao, Peng Wang, and Robert X. Gao. Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, 115:213–237, 2019.
- [33] albertbup. A python implementation of deep belief networks built upon numpy and tensorflow with scikit-learn compatibility, 2017.
- [34] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vish-

wanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.