

Article

Real-Time Remaining Useful Life Prediction of Cutting Tools Using Sparse Augmented Lagrangian Analysis and Gaussian Process Regression

Xiao Qin ^{1,†}, Weizhi Huang ^{2,†}, Xuefei Wang ³, Zezhi Tang ⁴ and Zepeng Liu ^{4,*} ¹ Adam Smith Business School, University of Glasgow, Glasgow G12 8QQ, UK² Department of Physics and Astronomy, University College London, London WC1E 6BT, UK³ School of Engineering, University of Manchester, Manchester M13 9PL, UK⁴ Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield S10 2TN, UK

* Correspondence: zepeng.liu@sheffield.ac.uk

† These authors contributed equally to this work.

Abstract: Remaining useful life (RUL) of cutting tools is concerned with cutting tool operational status prediction and damage prognosis. Most RUL prediction methods utilized different features collected from different sensors to predict the life of the tool. To increase the prediction accuracy, it is often necessary to mount a great deal of sensors on the machine in order to collect more types of signals, which can heavily increase the cost in industrial applications. To deal with this issue, this study, for the first time, proposed a new feature network dictionary, which can enlarge the number of candidate features under limited sensor conditions, and the developed dictionary can potentially contain as much useful information as possible. This process can replace the installation of more sensors and incorporate more information. Then, the sparse augmented Lagrangian (SAL) feature selection method is proposed to reduce the number of candidate features and select the most significant features. Finally, the selected features are input to the Gaussian Process Regression (GPR) model for the RUL estimation. Extensive experiments demonstrate that our proposed RUL estimation framework output performs traditional methods, especially for the cost savings for on-line RUL estimation.



Citation: Qin, X.; Huang, W.; Wang, X.; Tang, Z.; Liu, Z. Real-Time Remaining Useful Life Prediction of Cutting Tools Using Sparse Augmented Lagrangian Analysis and Gaussian Process Regression. *Sensors* **2023**, *23*, 413. <https://doi.org/10.3390/s23010413>

Academic Editor: Shen Zhang

Received: 30 November 2022

Revised: 22 December 2022

Accepted: 27 December 2022

Published: 30 December 2022



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: remaining useful life estimation; cutting tools; advanced manufacturing; sparse augmented lagrangian; gaussian process regression

1. Introduction

In recent years, the fourth industrial revolution (Industry 4.0) has brought about an increase in demand for precision machining and production manufacturing. As one of the typical manufacturing machines, high-speed computer numerically controlled (CNC) milling machines are widely used in industry [1]. Statistics show that CNC milling machines now account for approximately 46% or more of the total modern industrial machining cluster [2]. In order to ensure the safe operation of CNC machines and the quality of the products, cutting tools have a crucial role in the overall CNC machine system. However, cutting tools are usually operated in high-pressure and harsh environments, which can result in wear and tear on the tools. Some of the studies show that the failure rate of cutting tools is higher than 38% among all mechanical failures of CNC machines. When tool wear exceeds a certain threshold, the machined part no longer meets the machining requirements, which in turn causes losses to the plant. Drawing on these insights, there is an urgent need for researchers to develop efficient and cost-effective approaches to estimate the real-time remaining useful life (RUL) of tools in order to free up sufficient time to plan replacements and repairs, which can prevent potential unscheduled machine downtime and shorten the manufacturing cycle time of products [3,4].

At present, a variety of active achievements have been made to estimate the RUL predictions of cutting tools including direct and indirect techniques [5]. Direct methods basically use machine vision systems to capture the images of the cutting tools by means of digital cameras or optical sensors [6–8]. For example, Prasad et al. [9] used a pair of stereo images taken from two different locations to measure the depth of wear. Then the measured wear was used as test data to predict flank wear using a back propagation neural network. Lanzetta [10] measured and classified various categories of wear (e.g., flank wear, crater wear, fracture, and breakage) of cutting tool inserts using a vision sensor approach. The approach comprised a camera, an autofocus zoom lens, and different types of illumination. A resolution enhancement algorithm was used by the author which can enhance the resolution of the measured picture to 40 $\mu\text{m}/\text{pixel}$. The advantages of direct methods are that they can provide high identification accuracy for the tool wear measurement. However, they are essentially off-line techniques that often result in unnecessary machine downtime and increased costs due to lost productivity. To address this issue, indirect methods have been extensively investigated which are conducted by measuring parameters that are related to the condition of the cutting tool, i.e., acoustic, vibration, force, and sound [11–13]. Compared with direct approaches, indirect methods facilitate the processing of different types of signals and can easily be applied to on-line monitoring, and, therefore, have been a popular topic for estimating the RUL of cutting tools.

The application of indirect methods is based on two stages containing feature extraction and feature-based RUL estimation [1]. In regard to feature extraction, the significant features are extracted from the candidate feature dictionary developed from the collected signals in order to represent the status of the cutting tool. With respect to the feature-based RUL estimation, the relationship between the extracted features and the status of the cutting tool is established to predict the life of the cutting tool. For example, Liu et al. [14] utilized 14 signal features developed from the collected acoustic emission (AE) signals during the feature extraction stage and input the features to the support vector regression (SVR) for the life prediction during the RUL estimation stage. The results show that the method can achieve a prediction accuracy of up to 94.35%. Hu et al. [15] extracted time-domain features, such as mean, variance, and kurtosis; frequency-domain features, such as spectral kurtosis and spectral skewness; and time-frequency domain features, such as wavelet energy as the significant features; and the Long Short-term Memory (LSTM) network was applied to conduct tool wear prediction. Li et al. [16] utilized 16 features from the force signals during the feature extraction stage and applied a fuzzy neural network for the RUL estimating stage. Wang et al. [17] proposed a multisensory data fusion approach for the feature extraction and SVR for the RUL estimation. As can be seen from the aforementioned literature, to deliver the desired RUL estimation performance of indirect methods, the feature extraction that converts the raw signal into useful knowledge about the health status of cutting tools plays a fundamental role [1]. To extract significant features, the size of the candidate feature dictionary is required large enough to contain sufficient information about the tool state. Nevertheless, this process gives rise to a significant challenge which requires more sensors to be fitted on the machine which can heavily increase the cost.

To solve this issue, in the present study, a new feature network dictionary is proposed during the off-line training stage that enlarges the number of candidate features under limited sensor conditions, and the developed candidate features in the dictionary can potentially contain as much useful information as possible. After that, the sparse augmented Lagrangian (SAL) feature selection method is proposed during the feature extraction stage which can reduce the number of candidate features and select the most significant features [18]. The characteristics of SAL are that it utilizes Split Augmented Lagrangian Shrinkage Algorithm (SALSA) to produce some intermediate features with a sub-sampling technique, and then only the features with high selecting probability are chosen as the final selected features. Therefore, SAL has a very high sparsity and only a very few significant features can be selected. These selected features are often associated with some specific sensors, and only these specific sensors will be used for on-line monitoring. The

sensors that are not selected can be removed from the machine, which can save costs for on-line monitoring. Finally, the selected features are input to the Gaussian Process Regression (GPR) model for the RUL estimation. GPR is a non-parametric Bayesian regression method that performs well on small datasets and has the capability to produce uncertainty measurements on the predictions [19]. Promising results have demonstrated the effectiveness of the newly proposed RUL estimation framework including SAL and GPR has high prediction accuracy, demonstrating the advantage over many existing methods.

The main contributions of this article can therefore be summarized as follows.

- First, the feature network dictionary is proposed to extend the number of candidate features under limited sensor conditions in order to contain as much useful information as possible.
- Second, a SAL feature selection approach is proposed. It has very high sparsity and only the significant features can be selected.
- Finally, during the on-line monitoring, only the selected sensors will be used, and the sensors that are not selected can be potentially removed from the machine creating a favorable opportunity for cost savings in real-world industry applications.

The remainder of this paper is organized as follows. In Section 2, the theoretical background of the feature network dictionary, SALSA, SAL for feature selection, and GPR are discussed in detail. Then, Section 3 is dedicated to a description of the first case study and Section 4 shows the results of the second case study. Section 5 discusses the proposed method in detail. Finally, Section 6 concludes this study.

2. Theoretical Background

2.1. Feature Network Dictionary

During the machining process, different types of sensors are installed on the machine to monitor the cutting tool status. Suppose $x_{i,j}(t)$, $t = 1, \dots, T$, $i = 1, \dots, I$ and $j = 1, \dots, J$ is the measured raw signal responses from the i -th sensor output during the j -th cutting process, under which the tool wear is denoted as y_j . The signal features of $x_{i,j}(t)$ can be used to assess the tool wear severity and are given by

$$\mathbf{P}_{i,j} = [P_{i,j}^1, P_{i,j}^2, \dots, P_{i,j}^n], n = 1, \dots, N \quad (1)$$

and

$$P_{i,j}^n = f_{i,j}^n(x_{i,j}(t)) \quad (2)$$

where the notation n represents the n -th signal feature and N represents the number of signal features for i -th sensor output. $P_{i,j}^n$ is the generated feature and $\mathbf{P}_{i,j}$ represents the feature matrix enclosing N signal features. If the features from all I sensor outputs are combined in a single matrix, the signal feature dictionary during the j -th cutting process is given by [20]

$$\begin{aligned} \mathbf{P}_j &= [\mathbf{P}_{1,j}, \mathbf{P}_{2,j}, \dots, \mathbf{P}_{I,j}] \\ &= [P_{1,j}^1, \dots, P_{1,j}^N, P_{2,j}^1, \dots, P_{2,j}^N, \dots, P_{I,j}^1, \dots, P_{I,j}^N], j = 1, \dots, J \end{aligned} \quad (3)$$

From (3), there are a total of NI features in the signal feature dictionary. To extend the size of the dictionary under the I sensor outputs conditions in order to contain as much useful information as possible, the feature network dictionary is developed where the process of the extension of (3) is an iterative process. Support \mathbf{P}_j^l is the extended dictionary after the l -th extension iteration, and the original feature dictionary (3) can be written as

$$\begin{aligned} \mathbf{P}_j^0 &= \mathbf{P}_j = [P_{1,j}^1, \dots, P_{1,j}^N, P_{2,j}^1, \dots, P_{2,j}^N, \dots, P_{I,j}^1, \dots, P_{I,j}^N] \\ &= [a_1, a_2, \dots, a_\phi], \phi = 1, \dots, NI \end{aligned} \quad (4)$$

Define \mathbf{A}^1 as an intermediate matrix during the l -th iteration, which records the result of the matrix product, providing

$$\mathbf{A}^1 = \mathbf{P}_j^{0T} \times \mathbf{P}_j^0 = \begin{bmatrix} (a_1)^2 & a_1 a_2 & \dots & a_1 a_\phi \\ a_2 a_1 & (a_2)^2 & \dots & a_2 a_\phi \\ \vdots & \vdots & \ddots & \vdots \\ a_\phi a_1 & a_\phi a_1 & \dots & (a_\phi)^2 \end{bmatrix}, \quad (5)$$

The vectorization of the matrix \mathbf{A}^1 , denoted $\mathcal{V}\mathcal{E}\mathcal{C}(\mathbf{A}^1)$, is the $1 \times \phi^2$ row vector:

$$\mathcal{V}\mathcal{E}\mathcal{C}(\mathbf{A}^1) = [(a_1)^2, a_1 a_2, \dots, a_1 a_\phi, a_2 a_1, (a_2)^2, \dots, a_2 a_\phi, \dots, a_\phi a_1, a_\phi a_1, \dots, (a_\phi)^2] \quad (6)$$

Define \mathcal{D} as a process to delete the element which has the same expression of a vector, so $\mathcal{D}(\mathcal{V}\mathcal{E}\mathcal{C}(\mathbf{A}^1))$ can be expressed as

$$\mathcal{D}(\mathcal{V}\mathcal{E}\mathcal{C}(\mathbf{A}^1)) = [(a_1)^2, a_1 a_2, \dots, a_1 a_\phi, (a_2)^2, \dots, a_2 a_\phi, \dots, (a_\phi)^2] \quad (7)$$

where one of the same expression elements of \mathbf{A}^1 , such as $a_2 a_1$ and $a_1 a_2$, will be deleted. Based on the permutations and combinations formula [21], the number of elements in $\mathcal{D}(\mathcal{V}\mathcal{E}\mathcal{C}(\mathbf{A}^1))$ is $\frac{NI(NI+1)}{2!}$.

Then, the extended dictionary after the first iteration is

$$\begin{aligned} \mathbf{P}_j^1 &= [\mathbf{P}_j^0, \mathcal{D}(\mathcal{V}\mathcal{E}\mathcal{C}(\mathbf{A}^1))] \\ &= \underbrace{[a_1, a_2, \dots, a_\phi]}_{1 \times NI} \underbrace{[(a_1)^2, a_1 a_2, \dots, a_1 a_\phi, (a_2)^2, \dots, a_2 a_\phi, \dots, (a_\phi)^2]}_{1 \times \frac{NI(NI+1)}{2!}} \end{aligned} \quad (8)$$

where the size of the dictionary \mathbf{P}_j^1 extends to $1 \times (NI + \frac{NI(NI+1)}{2!})$.

In the same way, after the l -th extension with $l > 0$ and $l \in \mathbb{Z}^+$, the extended dictionary can be expressed as

$$\mathbf{P}_j^l = [\mathbf{P}_j^0, \mathcal{D}(\mathcal{V}\mathcal{E}\mathcal{C}(\mathbf{A}^l))] \quad (9)$$

where

$$\mathbf{A}^l = \mathbf{P}_j^{0T} \times \mathbf{P}_j^{l-1} \quad (10)$$

and the dictionary \mathbf{P}_j^l is defined as the feature network dictionary during the j -th cutting process having the size of $1 \times (NI + \binom{NI+1}{2} + \binom{NI+2}{3} + \dots + \binom{NI+l}{l+1})$ where $\binom{NI+l}{l+1} = \frac{(NI+l)!}{(l+1)!(NI-1)!}$. To facilitate understanding of the process of feature network dictionary generation, Figure 1 further demonstrates the procedure.

As a consequence, after a total of J cutting processes, the feature network dictionary can be rewritten as

$$\mathbf{P} = [\mathbf{P}_1^l, \dots, \mathbf{P}_j^l]^T \quad (11)$$

where the feature network dictionary \mathbf{P} is a $J \times (NI + \binom{NI+1}{2} + \binom{NI+2}{3} + \dots + \binom{NI+l}{l+1})$ matrix.

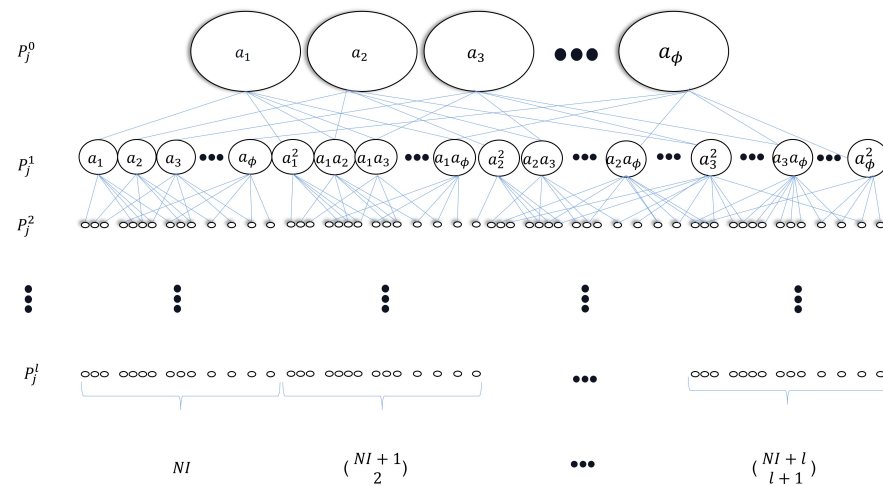


Figure 1. Process of generating the feature network dictionary during the j -th cutting process.

2.2. Split Augmented Lagrangian Shrinkage Algorithm (SALSA)

Assume that the tool wear vector is $\mathbf{y} = [y_1, \dots, y_J]^T$, then the tool wear estimation model can be expressed as

$$\mathbf{y} = \mathbf{P}\Theta + \Xi \tag{12}$$

where Θ stands for the model coefficients, and $\Xi = [\xi_1, \dots, \xi_J]^T$ is the unmodeled noise.

The feature network dictionary is required to be large enough to contain sufficient information with the aim of giving an accurate estimation of the tool wear. However, the features in the dictionary \mathbf{P} are usually redundant and may not be necessary for the tool wear estimation. A sparse model representation is therefore beneficial in this case [22,23].

The sparse solution of $\mathbf{y} = \mathbf{P}\Theta + \Xi$ can be obtained by solving the following l_1 -norm optimization problem [24]:

$$\Theta = \arg \min_{\Theta} \left\{ \frac{1}{2} \|\mathbf{P}\Theta - \mathbf{y}\|_2^2 + \lambda \|\Theta\|_1 \right\} \tag{13}$$

SALSA is applied to solve this equation, which is a variable splitting technique converting the original l_1 -norm minimization problem into a series of subproblems which can be solved separately. Equation (13) can be therefore converted as:

$$\begin{aligned} \min_{\Theta, v \in \mathbb{R}^M} \quad & f_1(\Theta) + f_2(v) + \frac{\mu}{2} \|\Theta - v\|_2^2 \\ \text{s.t.} \quad & v - \Theta = 0 \end{aligned} \tag{14}$$

where $f_1(\Theta) = \frac{1}{2} \|\mathbf{P}\Theta - \mathbf{y}\|_2^2$, $f_2(v) = \lambda \|v\|_1$ and μ is the Lagrange multiplier. The solution of (14) can be approximated to the weighted l_1 -norm optimization problem (13) with the increase of μ . By applying the augmented Lagrangian method, the optimization problem of (14) is then expressed as:

$$\begin{aligned} L_{\mu}(\Theta, v, u) = & f_1(\Theta) + f_2(v) - u^T(\Theta - v) \\ & + \frac{\mu}{2} \|\Theta - v\|_2^2 \end{aligned} \tag{15}$$

where u is a vector of Lagrange multipliers. Substitute $d = u/\mu$ into (15), then the problem is transformed into:

$$L_{\mu}(\Theta, v, d) = f_1(\Theta) + f_2(v) + \frac{\mu}{2} \|\Theta - v - d\|_2^2 \tag{16}$$

The problem (16) is then solved by converting it into three suboptimization problems: solving Θ , u and d individually, and in detail:

$$\hat{\Theta}_{k+1} = (\mathbf{P}^T \mathbf{P} + \mu \mathbf{I})^{-1} (\mathbf{P}^T \mathbf{y} + \mu(v_k + d_k)) \quad (17)$$

$$v_{k+1} = \max\left(0, \left(\hat{\Theta}_{k+1} - d_k\right) - \mu/\lambda\right) - \max\left(0, -\left(\hat{\Theta}_{k+1} - d_k\right) - \mu/\lambda\right) \quad (18)$$

$$d_{k+1} = d_k - \left(\hat{\Theta}_{k+1} - v_{k+1}\right) \quad (19)$$

where $\hat{\Theta}_{k+1}$ represents the estimation of Θ and λ is the penalty parameter. It can be seen that $\hat{\Theta}_{k+1}$ is calculated iteratively and the iteration will stop when the nonzero elements of $\hat{\Theta}_{k+1}$ and $\hat{\Theta}_k$ have the same sign and location. Algorithm 1 briefly summarizes the main procedure of the SALS algorithm.

Algorithm 1 SALS Algorithm

Input: Model signal \mathbf{y} and dictionary \mathbf{P}

Initialization: sets $k = 0$, the Lagrange multiplier $\mu = 0.1$, penalty parameter $\lambda = 1 \times 10^{-3}$,

$$v_0 = d_0 = 0$$

1: **while** sign $\hat{\Theta}_{k+1} \neq \hat{\Theta}_k$, $loc_{k+1} \neq loc_k$ **do**

$$2: \quad \hat{\Theta}_{k+1} = (\mathbf{P}^T \mathbf{P} + \mu \mathbf{I})^{-1} (\mathbf{P}^T \mathbf{y} + \mu(v_k + d_k))$$

$$3: \quad v_{k+1} = \max\left(0, \left(\hat{\Theta}_{k+1} - d_k\right) - \mu/\lambda\right) - \max\left(0, -\left(\hat{\Theta}_{k+1} - d_k\right) - \mu/\lambda\right)$$

$$4: \quad d_{k+1} = d_k - \left(\hat{\Theta}_{k+1} - v_{k+1}\right)$$

5: $k \leftarrow k + 1$

6: **end while**

2.3. Sparse Augmented Lagrangian (SAL) Algorithm for Feature Selection

By applying the SALS algorithm, the l_1 -norm optimization problem can be solved. However, the results are often not sufficiently sparse. To solve this issue, the SAL algorithm is used to improve the performance of SALS. Firstly, SAL uses SALS to generate a series of intermediate models. Secondly, certain features from the model series are selected to build the final model. The detailed procedure of the SAL algorithm is presented as follows:

- (1) The downsampling method is repeatedly used to process the tool wear vector \mathbf{y} and the feature network dictionary \mathbf{P} . Specifically, some rows are randomly selected from \mathbf{y} to form a new vector as \mathbf{y}_v . Similarly, the same rows are also selected from \mathbf{P} to form a new matrix denoted as \mathbf{P}_v . After that, the intermediate models based on \mathbf{y}_v and \mathbf{P}_v can be estimated as

$$\mathbf{y}_v = \mathbf{P}_v \Theta_v + \Xi_v \quad (20)$$

where v indicates the v -th repetition of the downsampling method with $v = 1, \dots, Y$, and $\mathbf{y}_v \in \mathbb{R}^{J_s}$, $\mathbf{P}_v \in \mathbb{R}^{J_s \times (NI + \binom{NI+1}{2} + \binom{NI+2}{3} + \dots + \binom{NI+I}{I+1})}$ and

$$J_s = \kappa \times J, 0 < \kappa < 1 \quad (21)$$

where J_s is the number of random subsampling, and κ represents the ratio of the downsampling.

- (2) SALS is then applied to calculate Θ_v from the v -th intermediate model using the downsampling data \mathbf{y}_v and \mathbf{P}_v .

- (3) The set of the selected features by SALSA for the v -th intermediate model can be expressed as

$$M_v = \{ \mathbf{p}^{\xi} | \Theta^{\xi} \neq 0 \}, \xi = 1, \dots, \left(NI + \binom{NI+1}{2} + \binom{NI+2}{3} + \dots + \binom{NI+l}{l+1} \right) \quad (22)$$

where $\mathbf{p}^{\xi} \in \mathbf{P}$ and $\Theta^{\xi} \in \Theta$. The notation ξ is the ξ_{th} element of the Θ , and \mathbf{p}^{ξ} are the ξ_{th} features.

- (4) Based on all $M_v, v = 1, \dots, Y$, the frequency of each feature being selected in all sets can be calculated. Define selection frequency (SF) as

$$\eta(\mathbf{p}_{\xi}) = \chi(\mathbf{p}_{\xi})/Y \quad (23)$$

where χ and η are the select times and the select frequency, respectively, of \mathbf{p}_{ξ} .

- (5) Then, the newly selected dictionary \mathbf{P}_s is defined as

$$\mathbf{P}_s = \{ \mathbf{p}_{\xi} | \eta(\mathbf{p}_{\xi}) \geq \delta \} \quad (24)$$

where the notation δ indicates a predefined feature selection threshold. This method can discard certain irrelevant features and only keep the features with high SF. The newly selected dictionary can be expressed as

$$\mathbf{P}_s = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{\theta}], \theta \leq \left(NI + \binom{NI+1}{2} + \binom{NI+2}{3} + \dots + \binom{NI+l}{l+1} \right) \quad (25)$$

- (6) Finally, the location of each selected feature in the feature network dictionary \mathbf{P} is recorded as Λ .

The final implementation of SAL for feature selection can be summarized in Algorithm 2 and Figure 2.

Algorithm 2 SAL for feature selection

Input: Model signal \mathbf{y} , dictionary \mathbf{P} , downsampling ratio κ and feature selection threshold δ

Output: Selected features \mathbf{P}_s and the location Λ .

- 1: **for** $v = 1 : Y$ **do**
 - 2: Random downsampling $\rightarrow \mathbf{y}_v$ and \mathbf{P}_v
 - 3: SALSA algorithm in **Algorithm 1**
 - 4: $M_v = \{ \mathbf{p}^{\xi} | \Theta^{\xi} \neq 0 \}$
 - 5: $\eta(\mathbf{p}_{\xi}) = \chi(\mathbf{p}_{\xi})/Y$
 - 6: $\mathbf{P}_s = \{ \mathbf{p}_{\xi} | \eta(\mathbf{p}_{\xi}) \geq \delta \} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{\theta}]$
 - 7: The location of each selected feature is denoted as Λ .
 - 8: **end for**
-

2.4. Gaussian Process Regression (GPR) for the Remain Useful Life (RUL) Estimation

After the features are selected by SAL, GPR is applied to conduct the RUL estimation of the cutting tool. Assume the selected features \mathbf{P}_s and the measured tool wear vector \mathbf{y} can be divided into two datasets which are the training datasets used for GPR model training and the validation/testing dataset used for model evaluation. The training datasets are denoted as \mathbf{P}_s^{α} and \mathbf{y}^{α} , respectively, where \mathbf{P}_s^{α} indicates the selected dictionary and \mathbf{y}^{α} indicates the measured tool wear vector. Similarly, the validation/testing datasets are denoted as \mathbf{P}_s^{β} and \mathbf{y}^{β} , respectively.

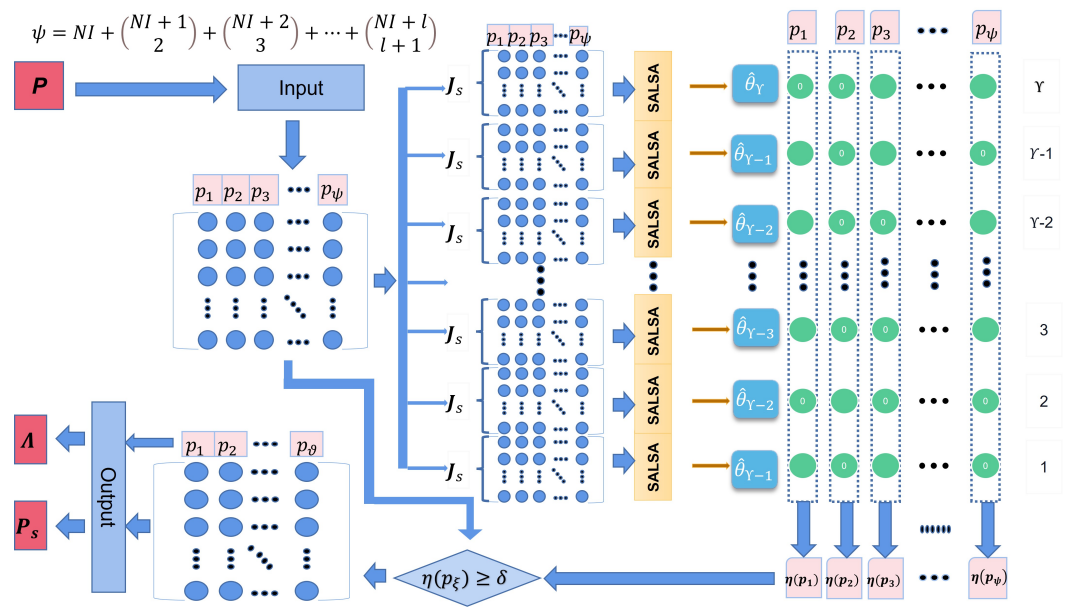


Figure 2. The implementation of SAL for feature selection.

The first step for the RUL estimation is to define a Gaussian kernel by using the training datasets. In the present study, the radial basis function (RBF) is selected as the kernel function because it has outstanding nonlinear ability and infinitely differentiable characteristics [19]. The RBF kernel is defined as [25]

$$K(\mathbf{P}_s^\alpha, \mathbf{P}_s^\alpha) = \sigma_f^2 \exp\left(-\frac{1}{2l} \|\mathbf{P}_s^\alpha - \mathbf{P}_s^\alpha\|\right) \tag{26}$$

where σ_f and l are hyperparameters.

After that, the predictive mean of the RUL of the validation/testing dataset denoted as $\bar{\mathbf{y}}^\beta$ can be derived

$$\bar{\mathbf{y}}^\beta = K(\mathbf{P}_s^\alpha, \mathbf{P}_s^\beta) \mathbf{L}^T \setminus (\mathbf{L} \setminus \mathbf{y}) \tag{27}$$

with

$$\mathbf{L} = \mathcal{C}(K(\mathbf{P}_s^\alpha, \mathbf{P}_s^\alpha) + \sigma_y^2 \mathbf{I}) \tag{28}$$

where

$$K(\mathbf{P}_s^\alpha, \mathbf{P}_s^\beta) = \sigma_f^2 \exp\left(-\frac{1}{2l} \|\mathbf{P}_s^\alpha - \mathbf{P}_s^\beta\|\right) \tag{29}$$

The function \mathcal{C} represents the Cholesky decomposition, and the notation σ_y presents the noise estimation parameter relating to the noise level of \mathbf{y} .

Then, the predictive variance of the RUL of the validation/testing dataset is represented as

$$\mathbb{V}(\mathbf{y}^\beta) = K(\mathbf{P}_s^\beta, \mathbf{P}_s^\beta) - \mathbf{v}^T \mathbf{v}, \mathbf{v} = \mathbf{L} \setminus K(\mathbf{P}_s^\alpha, \mathbf{P}_s^\beta) \tag{30}$$

where $\mathbb{V}(\cdot)$ indicates the variance function.

Finally, the log marginal likelihood can be calculated as

$$\log p(\mathbf{y} | \mathbf{P}_s^\alpha) = -\frac{1}{2} \mathbf{y}^T \mathbf{\Gamma} - \sum_i \log \mathbf{L}_{ii} - \frac{J}{2} \log 2\pi \tag{31}$$

where the notation $\mathbf{\Gamma} = \mathbf{L}^T \setminus (\mathbf{L} \setminus \mathbf{y})$.

2.5. RUL Estimation of the Cutting Tool

The cutting tool RUL estimation can be performed using the following algorithm, which has two parts, including off-line training and on-line RUL estimation.

Firstly, off-line training will extract the significant features from the proposed feature network dictionary, and train the GPR model off-line. The detailed process is as follows:

- (1) The collected raw signals from different sensor outputs are used to generate a feature network dictionary denoted as \mathbf{P} , and the features inside the dictionary \mathbf{P} are normalized by using min-max normalization.
- (2) The normalized feature network dictionary \mathbf{P} and the measured tool wear vector \mathbf{y} are divided into two parts which are the training datasets and validation datasets. The training datasets are denoted as \mathbf{P}^α and \mathbf{y}^α , respectively, and the validation datasets are denoted as \mathbf{P}^β and \mathbf{y}^β , respectively.
- (3) The generated dictionary \mathbf{P}^α and the measured tool wear vector \mathbf{y}^α are used to train the SAL feature selection model, and the selected dictionary is denoted as \mathbf{P}_s^α , where the location of each selected feature in the dictionary \mathbf{P}^α is recorded as Λ . Based on Λ , the significant features from the validation dataset denoted as \mathbf{P}_s^β can be directly selected.
- (4) GPR is conducted to estimate the RUL of the cutting tools. First, \mathbf{P}_s^α is applied to define the Gaussian kernel. Then, the predictive mean denoted as $\bar{\mathbf{y}}_s^\beta$ and the predictive variance denote as $\mathbb{V}(\mathbf{y}^\beta)$ of the validation dataset can be estimated.
- (5) Mean square error (MSE) shown below is used to evaluate the predictive results.

$$\text{MSE}^\beta = \frac{1}{\zeta} \left\| \mathbf{y}_s^\beta - \bar{\mathbf{y}}_s^\beta \right\|^2 \quad (32)$$

where the notation ζ indicates the length of \mathbf{y}_s^β . If the MSE^β is greater than a predefined threshold \mathcal{T}_β , the corresponding parameters of the SAL feature selection procedure need to be re-tuned, and (4) and (5) need to be repeated until $\text{MSE}^\beta \leq \mathcal{T}_\beta$.

The second part is the on-line RUL estimation using the on-line extracted features and off-line trained SAL and GPR models to conduct the on-line RUL prediction. The procedure is as follows:

- (1) Based on the location Λ , the significant features can be directly selected from the on-line collected signals.
- (2) The selected features are input to the off-line trained GPR model for the on-line RUL estimation.

Figure 3 presents a schematic flowchart of the proposed off-line training and on-line RUL estimation.

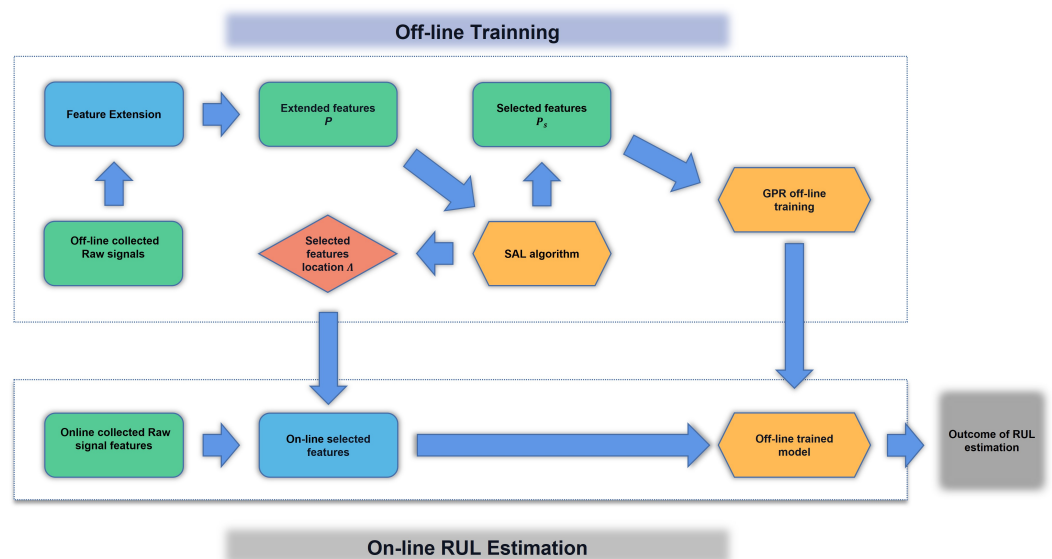


Figure 3. The process of off-line training and on-line RUL estimation of the Cutting Tool.

3. Case Study 1: PHM Data Challenge Datasets

3.1. Experiment Setup

To demonstrate the effectiveness of the proposed SAL and GPR framework, the down milling operation experiment conducted by the PHM society in 2010 is employed as shown in Figure 4 [26]. As can be seen, different types of sensors including the dynamometer, accelerometer, and acoustic emission (AE) sensor are used to mount on the different places of the machine with the aim of collecting sensor signals to conduct the RUL of the cutting tools. First, a Kistler quartz 3-component dynamometer is installed between the workpiece and the machining table in order to measure the cutting forces in X, Y, and Z directions during machining. The collected signals are denoted as $force_x$, $force_y$, and $force_z$, respectively. Then, three Kistler piezo accelerometers denoted as $acce_x$, $acce_y$, and $acce_z$, respectively, are mounted in the X, Y, and Z directions of the workpiece in order to measure the vibrations in each of these directions. Furthermore, a Kistler AE sensor denoted as AE_{RMS} is used, which can measure high-frequency energy signals generated during material removal from the workpiece in the machining process. As a consequence, seven sensor outputs which are $force_x$, $force_y$, $force_z$, $acce_x$, $acce_y$, $acce_z$, and AE_{RMS} , respectively, are collected from the applied sensors, and these sensor outputs are sampled by a NI DAQ PCI 1200 board with 12 kHz sampling rate.

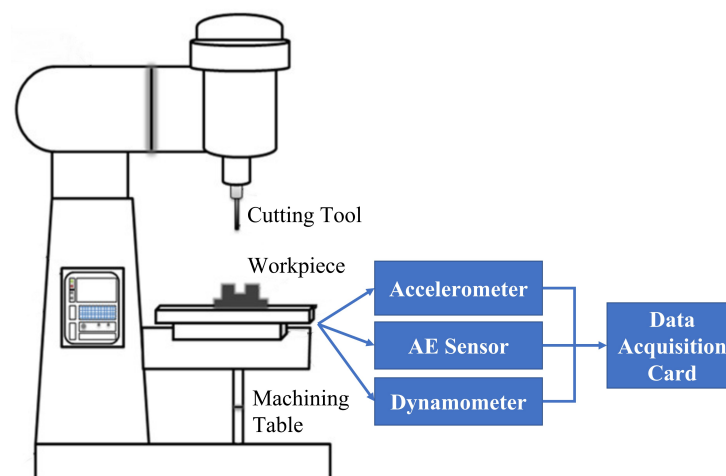


Figure 4. Illustration of the test rig.

During the experiment, the cutting tool with three flute cutters was utilized to cut the workpiece in order to machine a sloping surface [16]. The cutting speed is set up to 4.7 m/min and the spindle speed is set to 23,600 rpm. After each cutting process, the machine will stop to measure the flank wear of the cutter, which was conducted by employing the LEICA MZ12 microscopy system. At the end of the experiment, a total of three cutting tools denoted as T1, T2, and T3, respectively, were used to machine three aluminium workpieces, and each cutting tool made a total of 300 cutting processes so as to produce 300 datasets enclosing 7 columns sensor outputs and 1 column tool wear measurements.

3.2. Off-Line Training and Validation

3.2.1. Sparse Augmented Lagrangian (SAL) Algorithm for the Feature Selection

To conduct the SAL-based feature selection, as can be seen in Figure 5, the original datasets collected from T1 and T2 were first used as the training and validation datasets and the rest dataset collected from T3 was used as the testing dataset. As shown in Table 1, this experimental design for data partitioning is denoted as E1 where the training and validation datasets contain 600 ($=300 \times 2$) datasets and the testing dataset contains 300 datasets.

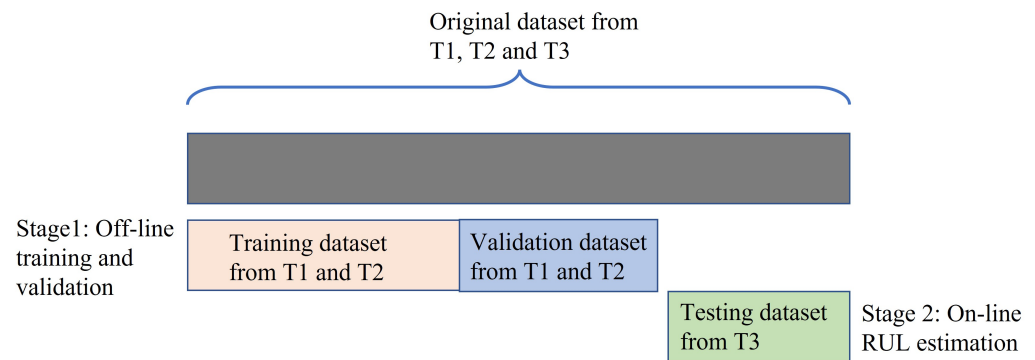


Figure 5. Design of the dataset partitioning of E1.

Table 1. Experimental design for data partitioning.

| Symbol | Training/Validation | Testing |
|--------|---------------------|---------|
| E1 | T1 and T2 | T3 |
| E2 | T1 and T3 | T2 |
| E3 | T2 and T3 | T1 |

During the off-line training and validation stage, 90% of the training and validation datasets (540 datasets) were randomly selected as the training dataset and the rest 10% of the datasets (60 datasets) were used as the validation dataset. Then, the training datasets were used to generate a feature network dictionary, and the features utilized in the current study are presented in Table 2. Based on the utilized features, a feature dictionary can be generated according to Equation (11). As can be seen in the equation, the critical parameter for the feature network dictionary is the extension iteration l . The ideal value of l can help in producing the gratifying size of the feature dictionary to contain as much useful information as possible. However, the increase of l will greatly expand the size of the initial dictionary matrix which heavily increases the computational load. For the current study, the initial extension iteration l is set to 1 for the SAL-based feature selection. According to Equation (11), the feature network dictionary of the training dataset is a 540×5670 matrix, meaning that the number of the designed features in the dictionary is extended to 5670. In a similar way, the validation dataset can also be used to generate a feature network dictionary where the size of the dictionary is 60×5670 .

After the feature dictionary is generated, the SAL algorithm was applied to select the significant features from the candidate 5670 features. Based on the algorithm introduced in Section 2.3, only 8 features denoted as \mathbf{P}_s^α were finally selected among the 5670 candidate features, where the selected features are AE_{RMS} , $force_x \times force_x$, $force_x \times force_y$, $force_x \times force_z$, $force_y \times force_y$, $force_y \times force_z$, $force_z \times force_z$, and $AE_{RMS} \times force_z$. The location of these selected features is recorded as Λ . As can be seen from the selected features, they are only related to the AE signal and the force signals, and there is no dependence on the vibration signals. Therefore, for the future on-line RUL estimation, there will be no need to install vibration sensors, which is very promising for industrial applications because the combination of feature network dictionary and SAL can be effective in cost saving. Finally, according to the location Λ of the selected features in the dictionary, the selected features of the validation dataset can be represented as \mathbf{P}_s^β .

Table 2. List of the utilized features.

| Features | Expression |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| mean | $\mu = \frac{1}{n} \sum_{i=1}^n x$ |
| variance | $\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x - \mu)^2$ |
| standard deviation | $\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x - \mu)^2}$ |
| skewness | $\gamma_1 = \frac{\frac{1}{n} \sum_{i=1}^n (x - \mu)^3}{\left(\frac{1}{n} \sum_{i=1}^n (x - \mu)^2\right)^{\frac{3}{2}}}$ |
| kurtosis | $K = \frac{\frac{1}{n} \sum_{i=1}^n (x - \mu)^4}{\left(\frac{1}{n} \sum_{i=1}^n (x - \mu)^2\right)^2}$ |
| root mean square | $X_{rms} = \sqrt{\frac{\sum_{i=1}^n x^2}{n}}$ |
| root mean square amplitude | $X_r = \left(\frac{1}{n} \sum_{i=1}^n \sqrt{ x }\right)^2$ |
| rectification average | $X_{arv} = \frac{1}{n} \sum_{i=1}^n x $ |
| peak to peak value | $X_p = x_{max} - x_{min}$ |
| waveform factor | $k_f = \frac{X_{rms}}{X_{arv}}$ |
| margin factor | $k_f = \frac{X_{max}}{X_{rms}}$ |
| peak factor | $k_a = \frac{x_p}{x_r}$ |
| median | $Q_{\frac{1}{2}}(x) = \begin{cases} x'_{\frac{n+1}{2}}, & n \text{ is odd} \\ \frac{x'_{\frac{1}{2}n} + x'_{\frac{1}{2}n+1}}{2}, & n \text{ is even} \end{cases}$ |
| maximum | $X_{max} = \max(x)$ |
| sum | $\mu = \sum_{i=1}^n x$ |

3.2.2. Gaussian Process Regression (GPR) for the RUL Estimation

After the features were selected via the SAL algorithm, they were then input to GPR for the RUL estimation. First, based on the algorithm introduced in Section 2.4, 8 selected features \mathbf{P}_s^α from the training dataset were used to train the GPR model. After that, the 8 selected features \mathbf{P}_s^β from the validation dataset were input to the trained GPR model in order to predict the RUL of the validation dataset. To estimate the performance of the prediction results, the mean square error (MSE) and the area of uncertainty (AoU) are used. The equation of the MSE is computed as

$$\text{MSE} = \frac{1}{\zeta} \sum_{\zeta=1}^{\Xi} (y_{\zeta}^v - \hat{y}_{\zeta}^v)^2 \quad (33)$$

where y_{ζ}^{β} and \hat{y}_{ζ}^{β} are the measured tool wear of the validation dataset and the GPR predicted tool wear, respectively. The notation Ξ indicates the length of the predicted tool wear \hat{y}_{ζ}^{β} , and ζ indicates the index of the dataset. The smaller the MSE, the more accurate the model's predictions will be. Furthermore, the area of uncertainty (AoU) is proposed to measure the uncertainty of the prediction results, which can be expressed as

$$\text{AoU} = \frac{1}{\Xi} \sum_{\zeta=1}^{\Xi} (4 \times \sigma_{\zeta}) \quad (34)$$

where the notation $\sigma_{\zeta} = \text{sqrt}(\mathbb{V}(\hat{y}_{\zeta}^{\beta}))$ represents the standard deviation of the prediction results at the ζ -th dataset. The smaller the AoU, the higher the reliability of the model's prediction will be. Figure 6a shows the validation result where the red dash line indicates the mean value of the estimation and the gray color represents the uncertainty of the estimation which is plotted by $\hat{y}_{\zeta}^v \pm 2 \times \sigma_{\zeta}$. As can be seen, the MSE is 39.84 and the AoU is 0.12.

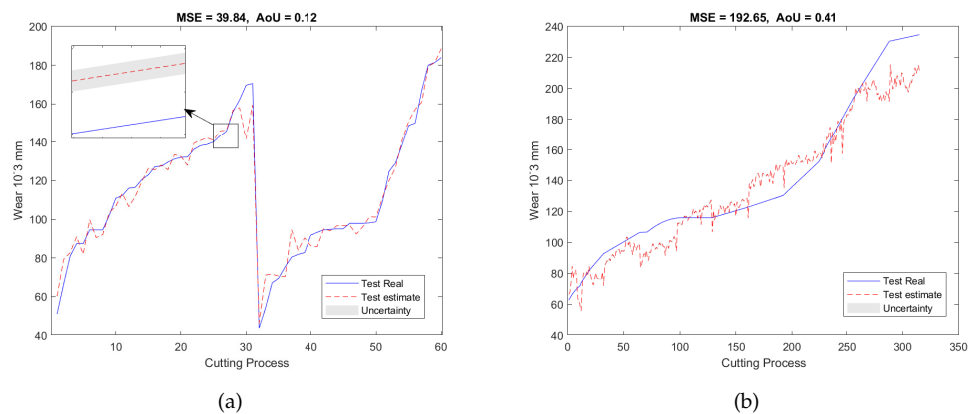


Figure 6. RUL prediction results of the experiment E1 of (a) the validation dataset and (b) the testing dataset.

3.3. On-Line RUL Estimation

After the SAL-based feature selection model and GPR model were trained by using the data from T1 and T2, they were then applied to the testing dataset acquired from T3 with the aim of conducting the on-line RUL estimation. First, the dataset from T3 was used to generate a feature network dictionary where the size of the dictionary is 300×5670 . Second, based on the off-line trained location Λ , the features of the test dataset were selected which can be expressed as $\mathbf{P}_s^{\beta^*}$. Finally, the selected features were input into the GPR model for the RUL estimation. Figure 6b shows the predicted RUL on the test dataset producing an MSE of 192.65 and an AoU of 0.41.

3.4. Additional Experiment Designs

To validate our proposed RUL estimation framework, as can be seen in Table 1, two additional experimental designs denoted as E2 and E3 were carried out. In the case of E2, the training and validation datasets are from T1 and T3, and the testing dataset is from T2; while for E3, the datasets acquired from T2 and T3 were used as the training and validation datasets, and the dataset collected from T1 was utilized as the testing dataset. For both experiments, 7 features were selected from 5670 candidate features during the training and validation stage. Similar to E1, the selected features are also associated with AE and force signals only, with no dependence on vibration signals. The validation results of E2 and E3 were presented in Figures 7a and 8a, respectively. Finally, after applying testing datasets to the trained SAL and GPR model, the RUL testing results are presented in Figures 7b and 8b. Tables 3 and 4 summarized the validation and testing results of E1, E2, and E3.

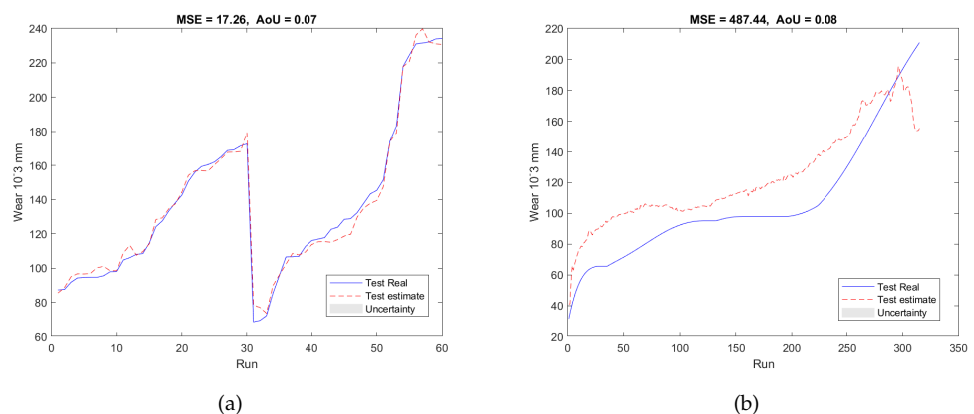


Figure 7. RUL prediction results of the experiment E2 of (a) the validation dataset and (b) the testing dataset.

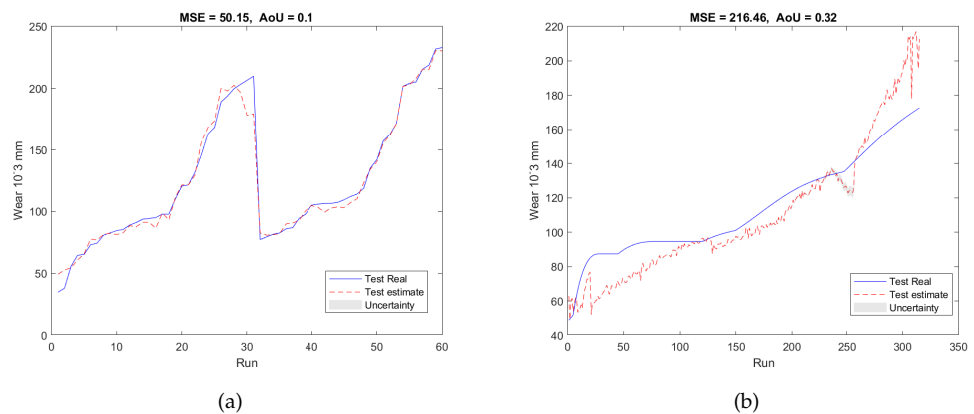


Figure 8. RUL prediction results of the experiment E3 of (a) the validation dataset and (b) the testing dataset.

Table 3. Summary of the validation results of E1, E2, and E3.

| Algorithms | MSE | | | AoU | | |
|-------------|--------------|--------------|--------------|-------------|-------------|-------------|
| | E1 | E2 | E3 | E1 | E2 | E3 |
| SAL and GPR | 39.84 | 17.26 | 50.15 | 0.12 | 0.07 | 0.10 |
| LASSO | 145.84 | 170.80 | 217.50 | Non | Non | Non |
| SAL | 123.45 | 89.74 | 67.31 | Non | Non | Non |
| GPR | 673.08 | 147.62 | 1192.31 | 0.12 | 5.15 | 0.08 |

Table 4. Summary of the testing results of E1, E2, and E3.

| Algorithms | MSE | | | AoU | | |
|-------------|---------------|---------------|---------------|-------------|-------------|-------------|
| | E1 | E2 | E3 | E1 | E2 | E3 |
| SAL and GPR | 192.65 | 487.44 | 216.46 | 0.41 | 0.08 | 0.32 |
| LASSO | 2103.42 | 419.67 | 279.72 | Non | Non | Non |
| SAL | 458.65 | 254.36 | 222.79 | Non | Non | Non |
| GPR | 372.16 | 202.71 | 313.83 | 0.15 | 5.18 | 0.08 |

Furthermore, the parameter l is also set to 0 to evaluate the RUL predictions in order to highlight the advantages of the feature extension in the feature network dictionary. For E1, based on Equation (11), with a non-extended dictionary, the number of features in the dictionary is only 105 which contains less useful information than the extended feature network dictionary. After applying SAL, the number of selected features is 39, which corresponds to the whole 7 sensor outputs. Compared to Figure 6, which utilizes the extended feature network dictionary, the prediction results of the non-extended dictionary (see Figure 9) show an inferior performance and use more sensors. In the same way, E2 and E3 also show poorer performance when $l = 0$. As a result, the extended feature network dictionary can potentially contain as much useful information as possible which can replace the installation of more sensors and incorporate more information meaning that the cost relating to sensor installation can be saved.

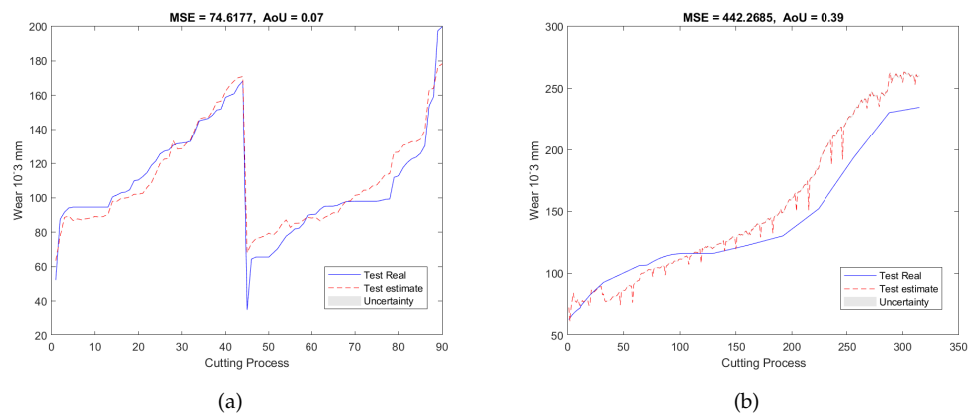


Figure 9. RUL prediction results of the experiment E1 under the non-extended dictionary condition ($l = 0$) of (a) the validation dataset and (b) the testing dataset.

3.5. Comparative Study

Since our method has two parts containing the SAL-based feature selection method and GPR prediction method, some comparisons are made in terms of RUL predictions by using a Least Absolute Shrinkage and Selection Operator (LASSO) approach [27], a pure SAL algorithm, and a pure GPR algorithm:

- First, with respect to the LASSO approach, as can be seen in Tables 3 and 4, the prediction results are promising and are close to that of our proposed method. However, the number of selected features exceeds 1000 in terms of E1, E2, and E3, which far exceeds the SAL-based feature selection method.
- Second, for the pure SAL algorithm, it can produce a very sparse solution with an adequate prediction performance (see Tables 3 and 4), but it cannot output the uncertainty of the model which limits the ability of engineers/researchers to have an evaluation of the reliability of the prediction results.
- Last, for the pure GPR method, as presented in Tables 3 and 4, it can output the uncertainty of the model, but the method is not sparse, i.e., it exploits the overall feature information for RUL prediction, which loses efficiency in high-dimensional spaces, especially when the number of features is tremendous.

As a result, the combination of SAL and GPR can take full advantage of the benefits of uncertainty estimation and overcome the drawbacks due to oversized features.

4. Case Study 2: NASA Ames Milling Datasets

To further demonstrate the effectiveness of the proposed RUL estimation framework, a second case study was conducted by analysing the NASA milling dataset [28]. Figure 10 depicts the setup of the experiment. As can be seen, AE sensors, vibration sensors, and motor current sensors are mounted to the table and the spindle of the Matsuura machining center MC-510V; moreover, six signal outputs which are represented as AE_{table} (AE_{table} indicates AE signals at table), $AE_{spindle}$ ($AE_{spindle}$ indicates AE signals at spindle), VB_{table} (VB_{table} indicates table vibration signals), $VB_{spindle}$ ($VB_{spindle}$ indicates spindle vibration signals), smc_{AC} (smc_{AC} indicates AC spindle motor current), and smc_{DC} (smc_{DC} indicates DC spindle motor current), respectively, are collected for estimating the RUL of the cutting tools. During the experiment, the cutting speed was set to 200 m/min, and two different depths of cut (DoC), namely, 1.5 mm and 0.75 mm, and two different feed rates, namely, 413 mm/min and 206.5 mm/min, were designed for the experiments. Furthermore, the workpieces for the experiments have two materials, which are cast iron and stainless steel J45 having the size of 483 mm × 178 mm × 51 mm. All experiments were performed a second time with the same parameters and a second cutting tool. The tools used for this case study are KC710-type carbide insets. Table 5 summarized the conducted experiments under different process parameter conditions. As can be seen from the table, it

contains 8 pairs of experiments with the same 8 different parameter settings. For example, Experiment 9 is a repeat experiment of Experiment 1. Therefore, the dataset acquired from Experiment 1 can be considered as the training/validation dataset and the dataset collected from Experiment 9 can be used as the testing dataset, and this pair of experiments can be denoted as P1. Similarly, the rest of the datasets can be named in the same way. Table 6 summarizes these experimental pairs.

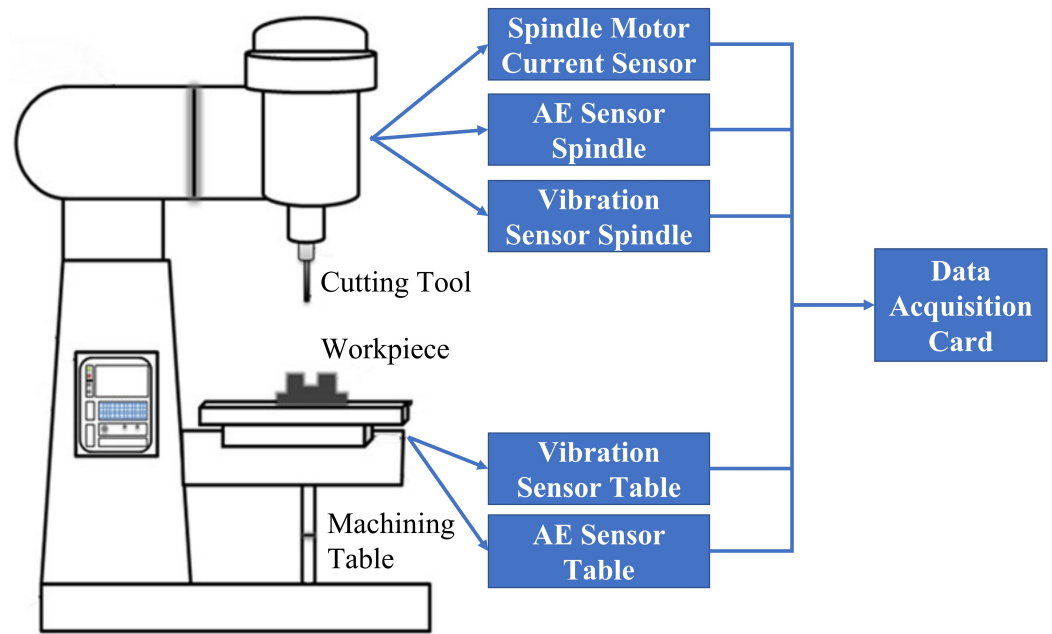


Figure 10. Experimental setup of the NASA Ames milling datasets [28].

Table 5. Experimental conditions of NASA milling datasets.

| Experiment | DoC | Feed | Material | Experiment | DoC | Feed | Material |
|------------|------|------|----------|------------|------|------|----------|
| 1 | 1.5 | 0.5 | castiron | 9 | 1.5 | 0.5 | castiron |
| 2 | 0.75 | 0.5 | castiron | 10 | 1.5 | 0.25 | castiron |
| 3 | 0.75 | 0.25 | castiron | 11 | 0.75 | 0.25 | castiron |
| 4 | 1.5 | 0.25 | castiron | 12 | 0.75 | 0.5 | castiron |
| 5 | 1.5 | 0.5 | steel | 13 | 0.75 | 0.25 | steel |
| 6 | 1.5 | 0.25 | steel | 14 | 0.75 | 0.5 | steel |
| 7 | 0.75 | 0.25 | steel | 15 | 1.5 | 0.25 | steel |
| 8 | 0.75 | 0.5 | steel | 16 | 1.5 | 0.5 | steel |

Table 6. Experimental pairs of NASA milling datasets.

| Symbol | Training/Validation | Testing |
|--------|---------------------|---------------|
| P1 | Experiment 1 | Experiment 9 |
| P2 | Experiment 2 | Experiment 12 |
| P3 | Experiment 3 | Experiment 11 |
| P4 | Experiment 4 | Experiment 10 |
| P5 | Experiment 5 | Experiment 16 |
| P6 | Experiment 6 | Experiment 15 |
| P7 | Experiment 7 | Experiment 13 |
| P8 | Experiment 8 | Experiment 14 |

Figure 11a shows the RUL estimation results of the validation dataset of the experiment pair P1. During the off-line training/validation stage, the number of the candidate features in the feature network dictionary is 4185, and only 9 features related to 3 sensors, namely, AE_{table} , $AE_{spindle}$, and $VB_{spindle}$, were selected. This process can result in considerable cost savings in industrial applications because unselected sensors will not be used during the on-line test. After that, for the on-line estimation, as can be seen in Figure 11b, the testing results are consistent with the validation results in terms of MSE and AoU, which implies that our proposed RUL estimation framework inherits some robustness. Furthermore, the traditional LASSO approach, SAL approach, and GPR approach were also used to analyze the dataset for comparison with our method. As can be seen from the pictures shown in Figures 12–14, they have higher MSE and AoU in comparison to our method. This means that the proposed framework has higher accuracy compared to other methods. Furthermore, Tables 7 and 8 present the validation results and testing results of the 8 experiment pairs. As can be seen, LASSO can have promising results on validations datasets but has poor results on testing datasets meaning that this method is not robust. In terms of the proposed SAL and GPR method, for the majority of cases of the testing dataset, the performance of the proposed method has higher accuracy than the traditional methods in terms of the MSE and AoU demonstrating the superiority of our proposed RUL estimation framework over traditional techniques.

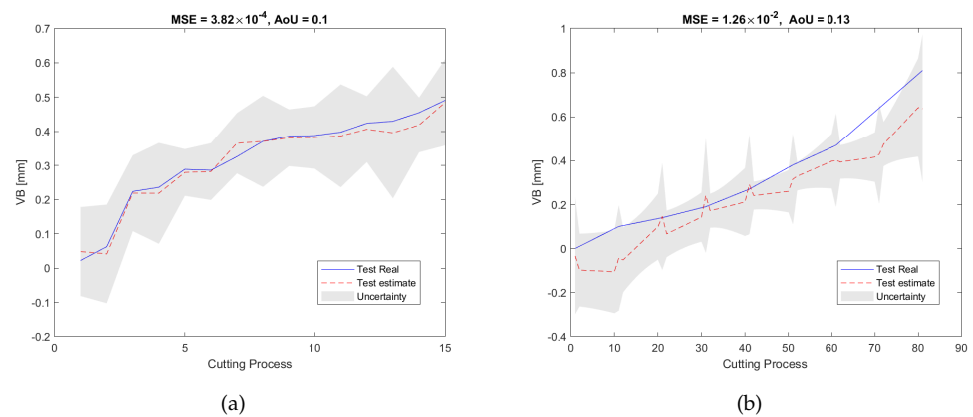


Figure 11. RUL prediction results of the proposed method of the experiment pair P1: (a) the validation dataset; (b) the testing dataset.

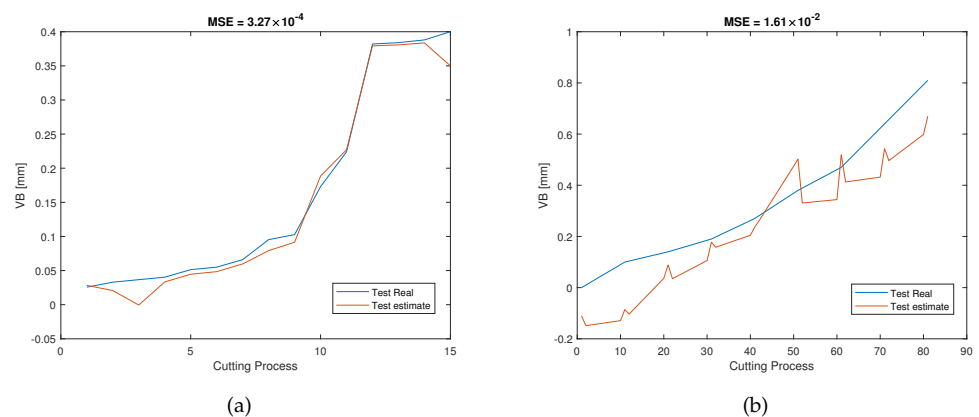


Figure 12. RUL prediction results of the LASSO method of the experiment pair P1: (a) the validation dataset; (b) the testing dataset.

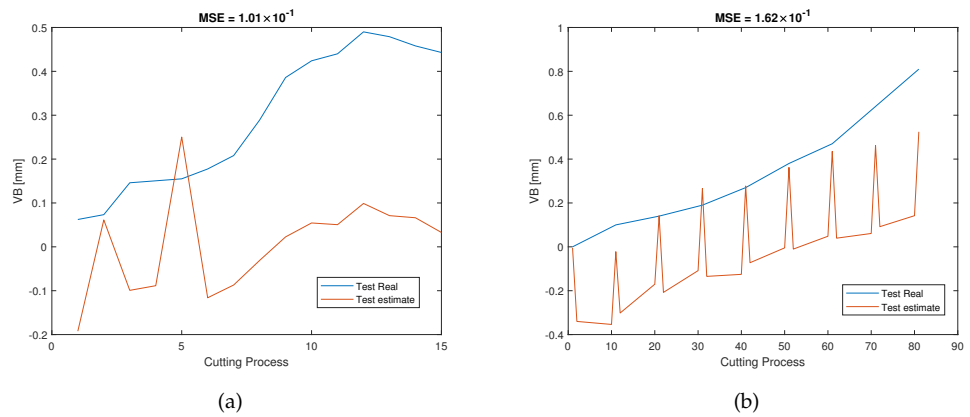


Figure 13. RUL prediction results of the SAL method of the experiment pair P1: (a) the validation dataset; (b) the testing dataset.

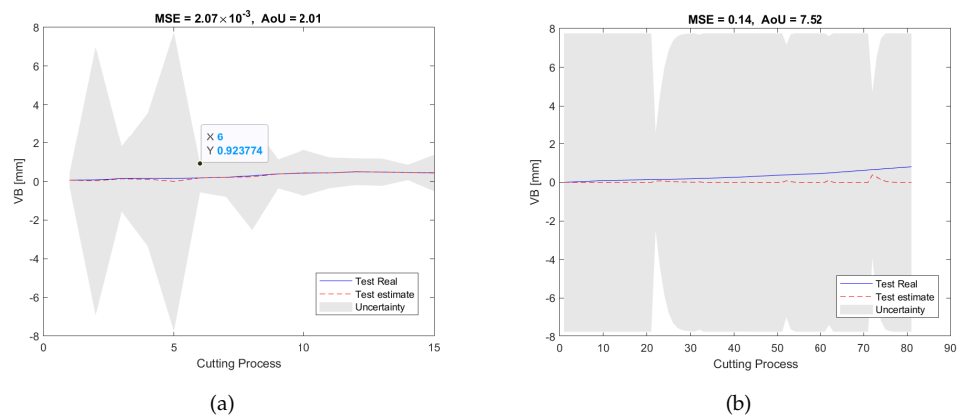


Figure 14. RUL prediction results of the GPR method of the experiment pair P1: (a) the validation dataset; (b) the testing dataset.

Table 7. Summary of the validation results of NASA milling datasets (a) MSE and (b) AoU.

| (a) | | | | | | | |
|-------------|----------------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Algorithms | MSE $\times 10^{-4}$ | | | | | | |
| | E1 | E2 | E3 | E4 | E5 | E7 | E8 |
| SAL and GPR | 3.82 | 179.00 | 8.03 | 26.10 | 49.70 | 5.34 | 43.00 |
| LASSO | 3.27 | 0.15 | 0.02 | 0.01 | 0.03 | 0.02 | 0.04 |
| SAL | 1010 | 189.00 | 45.00 | 8.00 | 5.00 | 40.00 | 25.00 |
| GPR | 20.70 | 195.82 | 594.62 | 7.61 | 4.88 | 63.82 | 5.74 |

| (b) | | | | | | | |
|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Algorithms | AoU | | | | | | |
| | E1 | E2 | E3 | E4 | E5 | E7 | E8 |
| SAL and GPR | 0.10 | 0.06 | 0.16 | 0.71 | 0.53 | 0.21 | 0.16 |
| LASSO | Non | Non | Non | Non | Non | Non | Non |
| SAL | Non | Non | Non | Non | Non | Non | Non |
| GPR | 2.01 | 0.06 | 0.17 | 1.87 | 1.52 | 2.64 | 1.73 |

Table 8. Summary of the testing results of NASA milling datasets, (a) MSE and (b) AoU.

| (a) | | | | | | | |
|-------------|----------------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Algorithms | MSE $\times 10^{-2}$ | | | | | | |
| | E1 | E2 | E3 | E4 | E5 | E7 | E8 |
| SAL and GPR | 1.26 | 2.14 | 3.36 | 0.74 | 1.02 | 31.31 | 3.08 |
| LASSO | 1.61 | 2.26 | 3.73 | 1.14 | 1.27 | 1.09 | 4.99 |
| SAL | 16.20 | 68.72 | 2.21 | 0.78 | 1.89 | 10.90 | 5.30 |
| GPR | 14.00 | 1.33 | 7.91 | 11.83 | 3.07 | 28.02 | 6.88 |
| (b) | | | | | | | |
| Algorithms | AoU | | | | | | |
| | E1 | E2 | E3 | E4 | E5 | E7 | E8 |
| SAL and GPR | 0.13 | 0.47 | 0.06 | 0.52 | 0.26 | 0.31 | 0.36 |
| LASSO | Non | Non | Non | Non | Non | Non | Non |
| SAL | Non | Non | Non | Non | Non | Non | Non |
| GPR | 7.52 | 0.06 | 0.08 | 3.00 | 0.13 | 0.71 | 1.19 |

5. Discussion

To better understand the advantages of the proposed SAL and GPR-based RUL estimation framework, some discussions are summarized as follows:

- (1) Sparsity: This study proposed a novel feature network dictionary which can extend the number of candidate features under limited sensor conditions with the aim of containing as much useful information as possible. However, the features in the dictionary are usually redundant and may not be necessary for the RUL estimation of cutting tools. The proposed SAL feature selection method can reduce the number of candidate features and select the most significant features. Compared with the conventional feature selection method, such as LASSO, SAL has a very high sparsity and only a very few significant features can be selected.
- (2) Cost saving: For traditional feature selection methods, they have low sparsity. Therefore, they lead to the selection of features associated with more sensors. In contrast, the high sparsity of SAL leads to the selection of fewer sensors, which can save costs for online monitoring.
- (3) Limitations: The process of feature network dictionary extension can enlarge the number of candidate features but increase a significant amount of computation to select the features. Therefore, a more efficient optimization algorithm is needed to reduce the computational requirements, which will be studied in the future research.

6. Conclusions

In this paper, an innovative feature network dictionary was proposed to enlarge the number of candidate features under limited sensor conditions, and the developed dictionary can potentially contain as much useful information as possible. As demonstrated by the experiments, this process can replace the installation of more sensors and incorporate more information. After that, the Sparse Augmented Lagrangian (SAL) feature selection method was applied to select the significant features and the Gaussian Process regression (GPR) algorithm was applied to estimate the Remaining Useful Life (RUL) of cutting tools. The results from several case studies demonstrated the effectiveness of the proposed technique.

This study can be treated as the initial research for RUL estimation of cutting tools because it mainly focuses on simple machining processes and strategies. In future research activities, more experimental cases including different cutting processes and machining strategies can be tested to validate our proposed methods. Furthermore, we plan to use different materials in the training and testing stages for evaluating the RUL prediction results.

Author Contributions: Z.L.: methodology; X.Q.: software; W.H.: validation; X.Q., W.H. and Z.T.: formal analysis; X.Q. and W.H.: investigation; X.Q. and W.H.: writing—original draft preparation; X.W., Z.T. and Z.L.: writing—review and editing; Z.L.: supervision; Z.L.: project administration. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Written informed consent has been obtained from the patient(s) to publish this paper.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, Z.; Lang, Z.Q.; Zhu, Y.P.; Gui, Y.; Laalej, H.; Stammers, J. Sensor Data Modeling and Model Frequency Analysis for Detecting Cutting Tool Anomalies in Machining. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, 1–13. [CrossRef]
2. Patange, A.D.; Jegadeeshwaran, R. Review on tool condition classification in milling: A machine learning approach. *Mater. Today Proc.* **2021**, 46, 1106–1115. [CrossRef]
3. Nasir, V.; Sassani, F. A review on deep learning in machining and tool monitoring: Methods, opportunities, and challenges. *Int. J. Adv. Manuf. Technol.* **2021**, 115, 2683–2709. [CrossRef]
4. Mohanraj, T.; Shankar, S.; Rajasekar, R.; Sakthivel, N.; Pramanik, A. Tool condition monitoring techniques in milling process—A review. *J. Mater. Res. Technol.* **2020**, 9, 1032–1042. [CrossRef]
5. Serin, G.; Sener, B.; Ozbayoglu, A.; Unver, H.O. Review of tool condition monitoring in machining and opportunities for deep learning. *Int. J. Adv. Manuf. Technol.* **2020**, 109, 953–974. [CrossRef]
6. Hou, Q.; Sun, J.; Huang, P. A novel algorithm for tool wear online inspection based on machine vision. *Int. J. Adv. Manuf. Technol.* **2019**, 101, 2415–2423. [CrossRef]
7. Ong, P.; Lee, W.K.; Lau, R.J.H. Tool condition monitoring in CNC end milling using wavelet neural network based on machine vision. *Int. J. Adv. Manuf. Technol.* **2019**, 104, 1369–1379. [CrossRef]
8. Peng, R.; Pang, H.; Jiang, H.; Hu, Y. Study of Tool Wear Monitoring Using Machine Vision. *Autom. Control. Comput. Sci.* **2020**, 54, 259–270. [CrossRef]
9. Dutta, S.; Kanwat, S.; Pal, S.; Sen, R. Correlation study of tool flank wear with machined surface texture in end milling. *Measurement* **2013**, 46, 4249–4260. [CrossRef]
10. Lanzetta, M. A new flexible high-resolution vision sensor for tool condition monitoring. *J. Mater. Process. Technol.* **2001**, 119, 73–82. [CrossRef]
11. Krishnakumar, P.; Rameshkumar, K.; Ramachandran, K. Acoustic emission-based tool condition classification in a precision high-speed machining of titanium alloy: A machine learning approach. *Int. J. Comput. Intell. Appl.* **2018**, 17, 1850017. [CrossRef]
12. Shi, X.; Wang, X.; Jiao, L.; Wang, Z.; Yan, P.; Gao, S. A real-time tool failure monitoring system based on cutting force analysis. *Int. J. Adv. Manuf. Technol.* **2018**, 95, 2567–2583. [CrossRef]
13. Arendra, A. Pre-processing for vibration signals features extraction and selection in real time investigating of CNC tool wear. *J. Phys. Conf. Ser. IOP Publ.* **2020**, 1569, 032060. [CrossRef]
14. Liu, Y.; Hu, X.; Sun, S. Remaining useful life prediction of cutting tools based on support vector regression. In Proceedings of the IOP Conference Series: Materials Science and Engineering, Wuhan, China, 10–12 October 2019; Volume 576, p. 012021.
15. Cai, W.; Zhang, W.; Hu, X.; Liu, Y. A hybrid information model based on long short-term memory network for tool condition monitoring. *J. Intell. Manuf.* **2020**, 31, 1497–1510. [CrossRef]
16. Li, X.; Lim, B.; Zhou, J.; Huang, S.; Phua, S.; Shaw, K.; Er, M. Fuzzy neural network modelling for tool wear estimation in dry milling operation. In Proceedings of the Annual Conference of the PHM Society, 2009; Volume 1. Available online: <https://papers.phmsociety.org/index.php/phmconf/issue/view/35> (accessed on 12 December 2022).
17. Wang, J.; Xie, J.; Zhao, R.; Zhang, L.; Duan, L. Multisensory fusion based virtual tool wear sensing for ubiquitous manufacturing. *Robot. Comput. Integr. Manuf.* **2017**, 45, 47–58. [CrossRef]
18. Tang, X.; Zhang, L.; Wang, X. Sparse augmented lagrangian algorithm for system identification. *Neurocomputing* **2019**, 330, 403–411. [CrossRef]
19. Wang, J. An intuitive tutorial to Gaussian processes regression. *arXiv* **2020**, arXiv:2009.10862.
20. Wang, X.; Liu, Z.; Lu, E. Remaining Useful Life Estimation of Cutting Tools Using Bayesian Augmented Lagrangian Algorithm. In Proceedings of the 2022 IEEE 31st International Symposium on Industrial Electronics (ISIE), Anchorage, AK, USA, 1–3 June 2022; pp. 1165–1169.
21. Velleman, D.; Call, G. Permutations and combination locks. *Math. Mag.* **1995**, 68, 243–253. [CrossRef]

22. Liu, Z.; Tang, X.; Wang, X.; Mugica, J.E.; Zhang, L. Wind Turbine Blade Bearing Fault Diagnosis Under Fluctuating Speed Operations via Bayesian Augmented Lagrangian Analysis. *IEEE Trans. Ind. Inform.* **2020**, *17*, 4613–4623. [[CrossRef](#)]
23. Liu, Z.; Yang, B.; Wang, X.; Zhang, L. Acoustic Emission Analysis for Wind Turbine Blade Bearing Fault Detection Under Time-Varying Low-Speed and Heavy Blade Load Conditions. *IEEE Trans. Ind. Appl.* **2021**, *57*, 2791–2800. [[CrossRef](#)]
24. Tang, X.; Zhang, L.; Li, X. Bayesian augmented Lagrangian algorithm for system identification. *Syst. Control. Lett.* **2018**, *120*, 9–16. [[CrossRef](#)]
25. Seeger, M. Gaussian processes for machine learning. *Int. J. Neural Syst.* **2004**, *14*, 69–106. [[CrossRef](#)] [[PubMed](#)]
26. PHM Society 2010 PHM Society Conference Data Challenge. Available online: https://phmsociety.org/phm_competition/2010-phm-society-conference-data-challenge/ (accessed on 12 December 2022).
27. Ranstam, J.; Cook, J. LASSO regression. *J. Br. Surg.* **2018**, *105*, 1348. [[CrossRef](#)]
28. Tyagi, V. NASA Milling Dataset. 2020. Available online: <https://www.kaggle.com/datasets/vinayak123tyagi/milling-data-set-prognostic-data> (accessed on 12 December 2022).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.