A Data Processing Pipeline for Prediction of Milling Machine Tool Condition from Raw Sensor Data

M. Ferguson[1], R. Bhinge[2], J. Park[3], Y. T. Lee[4], and K. H. Law[1]

Abstract

With recent advances in sensor and computing technology, it is now possible to use real-time machine learning techniques to monitor the state of manufacturing machines. However, making accurate predictions from raw sensor data is still a difficult challenge. In this paper, we describe how a data processing pipeline is developed to predict the condition of a milling machine tool using raw sensor data. Acceleration and audio time series sensor data is aggregated into blocks that correspond to the individual cutting operations of the Computer Numerical Control (CNC) milling machine. Each block of data is preprocessed using well-known and computationally efficient signal processing techniques. A novel kernel function is proposed to approximate the covariance between preprocessed blocks of time series data. Several Gaussian process regression models are trained to predict tool condition, each with a different covariance kernel function. The model with the novel covariance function outperforms the models that use more common covariance functions. The trained models are expressed using the Predictive Model Markup Language (PMML), where possible, to demonstrate how the predictive model component of the pipeline can be represented in a standardized form. The tool condition model is shown to be accurate, especially when predicting the condition of lightly worn tools.

[1] Stanford University, Civil and Environmental Engineering, Stanford, CA, USA
[2] Infinite Uptime, Inc. Berkeley, CA, USA
[3] Korea Advanced Institute of Science and Technology, Industrial and Systems Engineering, Daejeon, South Korea
[4] National Institute of Standards and Technology, Systems Integration Division, Gaithersburg, MD, USA

1 Introduction

The adoption of predictive models within the industrial sector promises to bring substantially increased operational effectiveness as well as the development of new services and products [1]. With the increasing availability of low-cost sensors, it is possible to collect real-time vibration and audio data from critical locations inside automated manufacturing machines. However, converting raw sensor data into accurate and actionable information is still a challenge. While the adoption of automated manufacturing machines has led to more efficient manufacturing processes, there is still a need for improved monitoring solutions [2].

For milling and turning processes, it is common for a human operator to use sound, vibration or visual methods to identify whether the mechanical process is operating smoothly. Replicating such monitoring techniques with computer systems, however, has proven difficult because of the volume, sparsity and noise that is associated with sensor data [3]. With the adoption of Computer Numerical Control (CNC) machines and production-line manufacturing, manufacturing machines operate with less human input. However, many automated manufacturing machines are unable to detect when a process is faulty until damage has been caused to the product or the machine. There is a need for better automated techniques to monitor manufacturing machines and to ensure that they operate efficiently.

In this work, we propose a technique for using acceleration and audio data to deduce the condition of a milling machine tool. In particular, we develop a pipeline for mapping raw sensor data to predicted tool condition values. Throughout the paper, we present solutions to real-time monitoring challenges: high data volume, high noise content, and information sparsity [3]. In particular, we show how the data volume challenge can be overcome by preprocessing the data into blocks. We then demonstrate that the data sparsity challenge can be overcome by using several well-established signal processing techniques, including the Fast Fourier Transform (FFT). Finally, we use the Gaussian Process Regression (GPR) technique to learn a mapping between the preprocessed sensor data and physical condition of the machine tool, in a manner that is not sensitive to noise in the observed condition.

Researchers have previously demonstrated that the condition of the machine tool can be inferred from features of the vibration and audio time series [4, 5]. However, we are yet to see widespread adoption of advanced monitoring techniques in the manufacturing domain, despite ongoing research in the field. Many of the existing models only work well for specific tasks, under tightly controlled laboratory conditions. A number of researchers have attempted to use the skew and kurtosis coefficients of the audio and acceleration time series to predict the condition of the tool, but with mixed results [5–7]. A wide range of tool monitoring techniques utilizing neural networks has also been reviewed by Dimla, Lister, and Leighton [8]. They concluded that artificial neural networks perform well on carefully selected experimental data, but there is a need for a multi-level system capable of handling unprocessed data. In particular, a useful tool condition monitoring system must automate the pre-processing, featurization and prediction steps, so the system can operate directly on raw sensor signals.

The remainder of this paper is organized as follows: In Section 2, we introduce the experimental setup and wireless data collection hardware. In Section 3, we provide an overview of the data processing pipeline and predictive model. In Section 4, we use several signal processing techniques to reduce the dimensionality and sparsity of the data. In Section 5, we describe how a Gaussian process regression model is developed to predict the tool condition of the CNC machine. The model is trained on the experimental data and the results are presented in Section 6. In Section 7, we discuss using the standardized Predictive Model Markup Language (PMML) to represent the predictive model. The paper is concluded with a summary and discussion.

2 Experimental Design and Monitoring Hardware

In this section, we describe how time series acceleration and acoustic data are collected from a CNC milling machine, namely a Mori Seiki NVD1500DCG. As shown in Fig. 1, a waterproof sensor unit from Infinite Uptime is attached to the vise of the milling machine. The sensor unit is capable of measuring both the audio and triaxial acceleration signals inside the milling machine. The acceleration signal is recorded in the x-, y- and z-directions. The acceleration sampling rate must be high enough to capture the signal from the four tool flutes as they remove material from the part. An acceleration sample rate of 1000 Hz is chosen to capture the 200 Hz signal generated by the cutting tool when the spindle speed is set to 3000 RPM. The audio signal is recorded at 8000 Hz. Data is streamed from the sensor to a laptop computer using a Universal Serial Bus (USB) connection.

The machining data, such as tool position and rotation speed, are recorded from the FANUC controller. An MTConnect [9] agent is used to synchronize the data from the milling machine and stream it to a laptop computer, along with a timestamp. A post-processing step is used to convert the machining data into a set of cutting operations performed by the machine. In the post-processing step, a simulation of the milling machine operations is used to distinguish actual material removal operations from other tool movements (e.g., movement above the workpiece) [10]. Using simulation, it is possible extract the behavior of the machine from the raw numerical control (NC) code. The details of the post-processing simulation step have been described in an earlier work [11].

The milling machine was programmed to produce a number of simple parts by removing material from a solid steel block. The production of each part required the milling machine to perform 20 separate cutting operations. The machine was instructed to produce parts until the cutting tool became severely damaged, or the cutting tool broke. Tests were conducted using an Atrax solid carbide 4-flute square end mill, and a continuous supply of coolant. The operating parameters of the machine were adjusted to increase the rate of tool wear in the experiments. The feed rate was increased and the rotation speed was decreased, thereby increasing the load on the cutting tool and making it wear faster. With the adjusted parameters, the milling tool life was reduced from several days to about 30 minutes. Even with the accelerated wear rate, we saw the same type of wear that is common under normal operation. However, it should be noted that, in some cases, we observed a rapid failure of the tool flutes, which is much more difficult to predict than gradual degradation. The operating parameters and experimental statistics are shown in Table 1.

A number of different methods have been proposed to measure the condition of a machine tool. Several authors investigating tool condition have used quantitative measurements, such as the wear depth, as tool condition measures. However, these measures often fail to fully describe the tool condition, especially when there is irregular wear or chipping in the tool flank [12]. In this work, we define the condition of the milling machine tool, $y \in [0,1]$, based on the remaining lifetime of the tool, as estimated after manually examining the tool with a microscope. The scale is defined such that 1.0 indicates a new tool in perfect condition, and 0.5 indicates the condition at which the tool would be replaced in a commercial manufacturing operation as judged by a machine operator. Fig. 2 shows examples of different levels of tool condition.

3 A Pipeline for Time Series Data Processing

One main contribution of this work is to provide a data processing pipeline that can be used to generate intermittent predictions from a high-volume stream of noisy time series data. Standardized data processing techniques and model representations are used where possible, to ensure that the data processing pipeline can easily be deployed in a real manufacturing facility. In this section, we provide an overview of the proposed data processing pipeline. In subsequent sections, we will describe each component of the data processing pipeline in more detail, using tool condition prediction as an example use-case.

Fig. 3 shows the proposed data processing pipeline. Acceleration and audio time series signals are recorded at the milling machine. In the first component of the pipeline, we use information from the CNC machine controller to label the blocks of time series data according to the machine action. This helps to organize the large time series signals into more manageable chunks. In the second component, the time series data is transformed into the frequency domain. This transformation allows the volume of the data to be reduced without significant loss of relevant information. We choose the FFT to create a frequency domain representation of the raw time series data, as it is computationally efficient and widely understood. In the final component, we use a trained predictive model to map the frequency-domain data to the output value, namely the tool condition.

The predictive model is represented using the standardized Predictive Model Markup Language (PMML) format [13]. This allows the trained model to be easily transferred from research to the factory floor, in a controlled and standardized manner. A scalable and production-ready implementation of this pipeline could be composed of a preprocessing module and a scoring module. The preprocessing module would reduce the time series data to feature vectors using FFT, and the scoring module would generate predictions using a PMML-compliant scoring engine. This pipeline would be relatively easy to implement in any computing environment that supports FFT and PMML.

4 Time Series Data Segmentation

In this section, we discuss how the time series data is segmented into blocks and labelled using the controller data. The rate of data produced by the acceleration and audio sensors is such that directly learning from the data is difficult. Furthermore, the acceleration and audio data is delivered as continuous stream of samples, while the actual manufacturing process is a series of more structured operations. When looking at the acceleration time series in Fig. 4, we observe that the amplitude of the acceleration is correlated with the type of cutting operation the machine is performing. A human operator monitoring the manufacturing machine would be aware of this correlation and use this supplementary information when analyzing the state of the machine. Therefore, segmenting the time series data according to machine operation is an important step for predicting tool condition.

The milling machine performs a number of different operations to produce a part. The data from the milling machine controller is used to automatically segment and label the time series data. Fig. 5 shows the time series data in the production of a part that involves 10 climb-cutting operations and 10 conventional-cutting operations. Each cutting operation is separated by a brief air-cutting operation. The term “air-cutting” is used to describe the operation of running the spindle without removing material. The terms “climb cutting” and “conventional cutting” refer to the relationship between the rotation direction to the feed direction. A detailed description of these operations is discussed in an earlier work [11].

4.1 AGGREGATION AND NOISE REDUCTION

As the audio and vibration signals are periodic, it seems natural to analyze the signals in the frequency domain. Technology involving filtering, modulation, and wave propagation often relies upon spectral analysis via, for example, the Fourier transform. A human machine operator relies on spectral analysis as well [14]. The machine operator classifies audio and vibration signals according to high and low pitch as well as purity, even if only subconsciously. In this section, we describe how the sparse and noisy time-domain signals are transformed into intermediate representations which capture the sort of information required to evaluate the state of a manufacturing process.

We assume that the frequency content of the recorded signals is solely dependent on the operation being performed by the machine and the condition of the machine tool. It follows from this assumption that the frequency content should be relatively constant over the duration of each machine operation. We also assume that the audio and acceleration signals are laden with noise. We attempt to deduce the power spectrum of the acceleration and audio signals, whilst reducing the influence of noise.

The discrete Fourier transform (DFT) is often used to estimate the power spectrum of a signal. However, the standard method of estimating the power spectrum using the Fourier transform tends to be susceptible to noise caused by imperfect and finite data [15]. To overcome some of these problems, we use an alternative technique introduced by Welch [16] that involves averaging multiple periodograms. Welch's method is an improvement on the standard periodogram method, in that it reduces noise in the estimated power spectra in exchange for reducing the frequency resolution [17]. In Welch’s method, a discrete time series signal s, is divided into K successive blocks s_m, using a window function w:

$$s_m(n) = w(n)s(n + mR), \quad n = 0 \dots M - 1, \qquad m = 0 \dots K - 1 \tag{1}$$

where M is the length of the window and R is the window hop size [18]. The parameters M and R control the number of periodograms which are averaged to give the periodogram estimate. The standard periodogram method uses a rectangular window, but we use the common Hann window function to reduce the spectral leakage [14]:

$$w(n) = \begin{cases} 0.5\left(1 - \cos\left(\frac{2\pi n}{M-1}\right)\right) & if \quad n \leq M-1; \\ 0 & otherwise. \end{cases} \tag{2}$$

The periodogram of the m^{th} block is calculated using the Fourier transform:

$$\boldsymbol{p}_m(\omega_k) = \frac{1}{M}\left|\sum_{n=0}^{N-1} s_m(n) e^{-\frac{i2\pi nk}{N}}\right|^2. \tag{3}$$

where ω_k is the kth point in the discretized frequency domain. The Welch estimate of the power spectral density is given by:

$$\hat{\boldsymbol{s}}(\omega_k) = \frac{1}{K}\sum_{m=0}^{K-1} \boldsymbol{p}_m(\omega_k). \tag{4}$$

Welch's method essentially computes the average of periodograms across time. Each periodogram is obtained from a windowed segment of the time series. In this study, a window overlap of 50% is used. A Hann window length of 256 points is chosen for the vibration and audio signals. The length of the window is equal to the number of intervals in the discretized frequency domain. We find that discretizing the frequency domain over 256 intervals allows us to form a minimal but sufficient representation of the true periodogram.

Alternative methods such as discrete wavelet transform (DWT) could be used to construct a time-frequency representation of the data in each time series block. However, in this work we are not interested in how the signal changes with time during a machine operation. Instead, we are interested in how the average signal generated during one machine operation compares with the average signal generated during a previous machine operation. Therefore, applying DFT with Welch's method is simpler and more suitable in this context.

When observing a milling machine operates in a lab setting, the acceleration signal amplitude increases as the cutting tool wears. As shown in Fig. 6, the acceleration signal amplitude is much larger when the tool is worn, across most frequencies. Another interesting observation is that the sharp tool produces a more peaked periodogram, corresponding to pure vibration modes, while the worn tool produces a flatter periodogram. Fig. 7 demonstrates that a worn tool produces an audio signal which tends to have a larger amplitude than that from a sharp tool.

5 Tool Condition Prediction with Gaussian Process Regression

In the previous section, we demonstrated how the acceleration and audio periodograms capture useful information about the tool condition. Next, we aim to develop a mapping between the periodograms and the tool condition by using the GPR technique, which provides a posterior distribution over the output by comparing input features using a kernel function. The regression task has a dataset $\mathcal{D} = \left\{\boldsymbol{x}^i, y^i\right\}_{i=1}^{n}$ with D-dimensional inputs $\boldsymbol{x}^i$ and scalar outputs y^i. We attempt to learn the unknown function $y^i = f(\boldsymbol{x}^i)$ by incorporating prior knowledge captured in the historical data.

In GPR, a Gaussian process (GP) is used as a prior to describe the distribution on the target function value $f(\boldsymbol{x})$ at an unseen input $\boldsymbol{x}$. A GP is a generalization of the Gaussian probability distribution for which any finite linear combination of samples has a joint Gaussian distribution [19]. A GP can be fully specified by its mean function $m(\boldsymbol{x})$ and covariance kernel function $k(\boldsymbol{x}, \boldsymbol{x}')$:

$$p(\boldsymbol{f}^{1:n}) = \mathcal{GP}\big(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')\big), \tag{5}$$

where $\boldsymbol{f}^{\boldsymbol{1:n}} = \big(f(x^1), f(x^2), \ldots, f(x^n)\big)$ are latent function values. The mean function and covariance kernel are defined as [20]:

$$m(\boldsymbol{x}) = \mathbb{E}[f(\boldsymbol{x})], \tag{6}$$

$$k(\boldsymbol{x}, \boldsymbol{x}') = \operatorname{cov}\big(f(\boldsymbol{x}), f(\boldsymbol{x}')\big). \tag{7}$$

The mean function $m(\boldsymbol{x})$ captures the overall trend in the target function value, and a kernel function $k(\boldsymbol{x}, \boldsymbol{x}')$ is used to approximate the covariance between the two function values $f(\boldsymbol{x})$ and $f(\boldsymbol{x}')$.

In the tool condition model, we choose to use the periodograms as input features. We denote the vibration periodogram $\hat{\boldsymbol{s}}_v^i \in \mathbb{R}^{256}$ and acoustic periodogram $\hat{\boldsymbol{s}}_a^i \in \mathbb{R}^{256}$ for each milling machine operation i. The condition of the milling tool tends to change gradually under normal operation. For this reason, we choose to include the previous condition of the milling tool in the feature vector. Altogether, we denote the input features for each milling operation i as:

$$\boldsymbol{x}^i = \begin{bmatrix} c^i \\ \hat{\boldsymbol{s}}_v^i \\ \hat{\boldsymbol{s}}_a^i \end{bmatrix}. \tag{8}$$

where c^i is the best estimate of the previous tool condition. Let the data from the production of each part be represented as the subset $\mathcal{D}_p = \{\boldsymbol{x}^k, y^k\}_{k=1}^{m}$, where m is the number of points recorded over the duration of the experiment. During the training procedure, the previous tool condition is known, so c^k can be set equal to y^{k-1}:

$$c^k_{training} = \begin{cases} 1 & for\ k = 1, \\ y^{k-1} & otherwise. \end{cases} \tag{9}$$

To make a new prediction using the scoring procedure, the feature vector $\boldsymbol{x}^{new}$ first needs to be derived from the milling machine data. In a manufacturing setting, the acceleration and audio periodograms can be calculated immediately after each operation is performed by the milling machine. However, the previous tool condition value y^{k-1}, will not be known. Therefore, we use the previous tool condition prediction, $\hat{y}^{k-1}$, in the scoring procedure, that is:

$$c^k_{testing} = \begin{cases} 1 & for\ k = 1, \\ \hat{y}^{k-1} & otherwise. \end{cases} \tag{10}$$

This makes the prediction process recursive. The first prediction is made by assuming a new tool, i.e. $c^k_{testing} = 1$. All subsequent predictions are made using the previous prediction as a starting point.

5.1 NOISE MODEL

Each tool condition label y^i in the training and testing datasets are assigned manually, so they are likely to contain some random error. We choose to model this error as random noise in the output value. Each observed value is modelled by the function $y^i = f(\boldsymbol{x}^i) + \epsilon$ where ϵ is the noise term. We assume that this noise follows an independent, identically distributed Gaussian distribution with zero mean and variance σ_ϵ^2:

$$\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2). \tag{11}$$

5.2 KERNEL FUNCTION

The covariance kernel function provides an efficient method to estimate the correlation between two feature vectors. In GPR, the kernel function is used to estimate the covariance between two input vectors, $\boldsymbol{x}^i$ and $\boldsymbol{x}^j$. The type of kernel function chosen can strongly affect the representability of the GPR model, and influence the accuracy of the predictions. The *squared exponential* (SE) kernel is a common choice for a GPR model:

$$k_{SE}(\boldsymbol{x}^i, \boldsymbol{x}^j) = \sigma^2 \exp\left(-\frac{1}{2\ell^2}\left|\left|\boldsymbol{x}^i - \boldsymbol{x}^j\right|\right|^2\right), \tag{12}$$

where the kernel function is described by the hyperparameters, σ and ℓ. The hyperparameter ℓ is commonly referred to as the length scale. The length scale quantifies whether two points at a certain distance apart in the input space are considered close together [19]. The signal variance σ^2

quantifies the overall magnitude of the covariance value. While the squared exponential (SE) kernel function is a good choice for many applications, it does not allow the length scale to vary for each dimension in the feature vector. A common solution is to use an *automatic relevance determination* (ARD) kernel, which assigns a different length scale to each dimension [21]. The ARD SE kernel is commonly used with GPR:

$$k_{ARD\ SE}(\boldsymbol{x}^i, \boldsymbol{x}^j) = \sigma^2 \exp\left(-\frac{1}{2}(\boldsymbol{x}^i - \boldsymbol{x}^j)^T \mathrm{diag}(\boldsymbol{\ell})^{-2}(\boldsymbol{x}^i - \boldsymbol{x}^j)\right). \tag{13}$$

An ARD SE kernel provides the flexibility to adjust the relevance (weight) of each parameter in the feature vector where the parameter vector $\boldsymbol{\ell} = (\ell_1, \dots \ell_i, \dots, \ell_m)$ quantifies the relevance of the input features. The ARD SE kernel has been shown to have good performance on a number of tasks including robot arm control and tool condition prediction [19, 22]. However, when using the ARD SE kernel, the number of hyperparameters grows linearly with the dimensional size of the feature vector. This can make the model prone to overfitting, depending on the hyperparameter optimization strategy [21].

An ideal kernel for the tool condition model would allow the length scale of each periodogram to be varied independently, without introducing a large number of additional hyperparameters. One common way to build a kernel over multiple data types is to add the kernels together [23]. We choose to create a new kernel by combining several Gaussian kernels. The resulting *sum of square exponential* (SSE) kernel is defined as follows:

$$\begin{aligned} k_{SSE}(\boldsymbol{x}^i, \boldsymbol{x}^j) = \sigma_1^2 \exp\left(-\frac{1}{2\ell_1^2}\left\|c^i - c^j\right\|^2\right) + \sigma_2^2 \exp\left(-\frac{1}{2\ell_2^2}\left\|\hat{\boldsymbol{s}}_v^i - \hat{\boldsymbol{s}}_v^j\right\|^2\right) \\ + \sigma_3^2 \exp\left(-\frac{1}{2\ell_3^2}\left\|\hat{\boldsymbol{s}}_{\boldsymbol{a}}^i - \hat{\boldsymbol{s}}_{\boldsymbol{a}}^j\right\|^2\right), \end{aligned} \tag{14}$$

where $\sigma_1, \sigma_2, \sigma_3, \ell_1, \ell_2$ and ℓ_3 are the parameters to be determined for the SSE kernel function. The first term in the SSE kernel captures the similarity of the previous tool condition, for two inputs $\boldsymbol{x}^i$ and $\boldsymbol{x}^j$. The second and third terms capture the similarity of the acceleration and audio periodograms, respectively. It is common practice in GPR model to include the noise term σ_ϵ^2 as another hyperparameter to be optimized in the training procedure. Altogether, a model with the SSE kernel function is described by seven hyperparameters, $\boldsymbol{\theta} = [\sigma_1, \sigma_2, \sigma_3, \ell_1, \ell_2, \ell_3, \sigma_\epsilon]$. When compared to the ARD kernel, the SSE kernel has fewer hyperparameters, but still allows the length scale of the previous state and periodograms to be adjusted independently.

5.3 TRAINING PROCEDURE

It follows from the assumption of Gaussian noise, that we can analytically infer a posterior distribution over Gaussian process functions, and analytically derive a marginal likelihood of the observed function values y given only hyperparameters $\boldsymbol{\theta}$, and the input vectors. The marginal distribution of the observations can be expressed as [21]:

$$p(\boldsymbol{y}^{1:n}|\boldsymbol{\theta}) = \int p(\boldsymbol{y}^{1:n}|\boldsymbol{f}^{1:n}, \boldsymbol{\theta})\, p(\boldsymbol{f}^{1:n}|\boldsymbol{\theta})\, d\boldsymbol{f}^{1:n}. \tag{15}$$

The unknown function $\boldsymbol{f}$ can be marginalized out of Eq 15 to obtain the marginal likelihood of the training observations. The hyperparameters $\boldsymbol{\theta}$ are chosen to maximize the marginal likelihood of the observations in the training dataset $\boldsymbol{\mathcal{D}}$. An optimization equation is formed to maximize the marginal likelihood, and obtain the optimum hyperparameters $\boldsymbol{\theta}^*$ [21]:

$$\boldsymbol{\theta}^* = \arg\max_{\theta} \log p\,(\mathbf{y}^{1:\mathrm{n}}|\boldsymbol{\theta}\,), \tag{16}$$

$$\boldsymbol{\theta}^* = \arg\max_{\theta}\left(-\frac{\mathbf{1}}{\mathbf{2}}(\mathbf{y}^{1:\mathrm{n}})^{\mathrm{T}}(\mathbf{K} + \sigma_\epsilon^2\mathbf{I})^{-\mathbf{1}}\mathbf{y}^{1:\mathrm{n}} - \frac{1}{2}\log|\mathbf{K} + \sigma_\epsilon^2\mathbf{I}| - \frac{\mathrm{n}}{2}\log 2\pi\right) \tag{17}$$

Finding the optimum hyperparameters using Eq 17 requires an iterative approach, as the value of the kernel matrix $\mathbf{K}$ is inherently dependent on the hyperparameters. The process for obtaining the optimum hyperparameters is well documented in the literature [21]. In this study, the MATLAB GPML (Gaussian Processes for Machine Learning) library [24] is chosen to optimize the hyperparameters.

5.4 SCORING PROCEDURE

In machine learning, the scoring process generally involves using a predictive model to estimate some output value based on a set of input features $\boldsymbol{x}$. In GPR, the aim of the scoring procedure is to return a posterior distribution f^{new} on the output value, based on the unseen input $\boldsymbol{x}^{new}$. In the case where the mean function is zero, the (hidden) response value $f^{new}$and the observed outputs $\boldsymbol{y}^{1:n} = \{y^1, \dots, y^n\}$ follow the joint Gaussian distribution as:

$$\begin{bmatrix} \boldsymbol{y}^{1:n} \\ f^{new} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} (\mathbf{K} + \sigma_\epsilon^2\mathbf{I}) & \boldsymbol{k} \\ \boldsymbol{k}^T & k(\boldsymbol{x}^{new}, \boldsymbol{x}^{new}) \end{bmatrix}\right), \tag{18}$$

where $\boldsymbol{k}^T = \left(k(\boldsymbol{x}^1, \boldsymbol{x}^{new}), \dots, k(\boldsymbol{x}^n, \boldsymbol{x}^{new})\right)$. The posterior distribution on the response f^{new} for the newly observed input $\boldsymbol{x}^{new}$ given the historical data can then be expressed as a Gaussian distribution:

$$f^{new} \sim \mathcal{N}\left(\mu(\boldsymbol{x}^{new}|\boldsymbol{\mathcal{D}}^n), \sigma^2(\boldsymbol{x}^{new}|\boldsymbol{\mathcal{D}}^n)\right). \tag{19}$$

The posterior mean $\mu(\boldsymbol{x}^{new}|\boldsymbol{\mathcal{D}}^n)$, and associated variance $\sigma^2(\boldsymbol{x}^{new}|\boldsymbol{\mathcal{D}}^n)$, can be calculated directly. That is, the mean and the variance of the posterior distribution can be expressed as [21]:

$$\mu(\boldsymbol{x}^{new}|\boldsymbol{\mathcal{D}}^n) = \boldsymbol{k}^T(\mathbf{K} + \sigma_\epsilon^2\mathbf{I})^{-1}\boldsymbol{y}^{1:n}, \tag{20}$$

$$\sigma^2(\boldsymbol{x}^{new}|\boldsymbol{\mathcal{D}}^n) = k(\boldsymbol{x}^{new}, \boldsymbol{x}^{new}) - \boldsymbol{k}^T(\mathbf{K} + \sigma_\epsilon^2\mathbf{I})^{-1}\boldsymbol{k}. \tag{21}$$

As the posterior distribution is one-dimensional Gaussian, the posterior mean and variance are sufficient to fully describe the posterior distribution.

6 Results

When evaluating machine learning models, it is common practice to divide the data set into a training set and a testing set. The model is trained on the training set and then tested on the previously unseen data in the testing set. In applied machine learning research, it is common to randomly assign data points to the training and testing sets, but this method is not recommended for time series data [25]. Instead, we randomly select 3 experiments for the testing set and assign data from the remaining 15 experiments to the training set.

Two GP models are trained to predict tool condition; the first is trained with climb-cutting data from the training set, and the second is trained with conventional-cutting data from the training set. The two models are then used together to predict the condition of the tool for each test case. Predictions are made in the order that the testing data was recorded.

In our experiments, models are trained using three different kernel functions. The prediction accuracy for each case is shown in Table 2. A model is said to generalize well if it achieves a similar performance on the training and testing set, or overfit if it fits well to the training data set but not to the test data set. The results demonstrate that the model with the SE kernel generalizes well, but it underfits the training data. The model with the ARD SE kernel however overfits the training data, but the model does not achieve good performance on the testing set. The model with the SSE kernel generalizes well and achieves reasonably good performance.

Fig. 8 shows the model prediction results when the previous tool condition state c^i is included in the input features. On the other hand, Fig. 9 shows tool condition predictions made without the state feature. The predictions in Fig. 9 show much larger variation in time than those with the previous state feature, as seen in Fig. 8. This qualitatively indicates that the previous condition state is a useful feature in the model.

Fig. 8 and Fig. 9 also reveal that the confidence interval on the predicted value is large when the tool is worn. There are several reasons for this: firstly, the tool is more likely to break or undergo rapid condition change at this stage; hence, there is less certainty when predicting the condition of a heavily worn tool. Secondly, the training set contains less data for heavily worn tools, as some of the tests were ended abruptly when the tool broke. The reduced amount of training data for heavily worn tools makes future predictions less certain. Finally, the amplitudes of the acceleration signal generated by heavily worn tools have a much larger variance. For example, if a single flute on the tool is worn heavily, the tool is no longer symmetric so the corresponding acceleration signal will have a very high amplitude. On the other hand, if the tool is worn evenly the corresponding acceleration signal may have a much smaller amplitude. Many of these limitations may have been exacerbated by the accelerated nature of the tool wear experiment. It is also important to note that tool condition would rarely be allowed to pass below 50 % in a real manufacturing setting. Once the tool exceeds this level of wear, the surface quality of the part deteriorates quickly.

7 Standardizing Components of the Data Processing Pipeline

When developing a predictive model for industrial use, it is highly beneficial if the trained model can be represented in a standardized manner [26]. The need for standardization is particularly high in large organizations, where the predictive models may be shared between people or departments. In this section, we discuss how the proposed data processing pipeline can be expressed in a standardized manner, using the PMML format [27].

The core of the proposed data processing pipeline is the GPR predictive model. The output of a trained GPR model is dependent on multiple factors, including the choice of mean kernel, the choice of covariance kernel, the model hyperparameters, and the training dataset. To transfer a trained GPR model between computing environments, it is also necessary to transfer this information. The PMML standard makes this information transfer less troublesome, by defining a standardized way to represent certain types of predictive models. A standardized representation for GPR models was introduced in the latest PMML specification, PMML 4.3 [28].

We express the GPR models developed in this work using PMML 4.3, where possible. The current PMML standard supports four covariance kernel types, namely the SE kernel, the ARD SE kernel, the absolute exponential kernel and the generalized exponential kernel [27]. Therefore, it is possible to represent the SE and ARD SE tool condition models with PMML 4.3. The PMML representation of a SE GPR model is shown in Fig. 10. The PMML representation of the ARD SE models is similar to that for the SE model, but it specifies a different covariance kernel and a different set of hyperparameters. The PMML 4.3 standard does not support covariance composite kernels that are created by the addition or multiplication of other standard kernel types. Unfortunately, the proposed SSE kernel cannot be represented using PMML 4.3. As addition or multiplication of kernels is commonly used to create new kernels with a range desirable properties [23], future versions of PMML should consider supporting the addition and multiplication of covariance kernels.

8 Summary and Discussion

In this study, we demonstrate how a data processing pipeline can be used to map raw sensor data to tool condition predictions. Information from the milling machine controller is used to divide the time series signals into discrete blocks of data. Each block of data is preprocessed using well-known and computationally efficient signal processing techniques. A non-parametric regression model, namely GPR, is then used to approximate the complex relationship between the preprocessed data and the target value. This pipeline-based approach is practical for use in a real manufacturing setting, as it allows raw sensor data to be mapped to tool condition predictions in a fast and robust manner.

The GP model provides confidence bounds for the predictive estimations, which are useful when interpreting the reliability of a prediction at some arbitrary time. The development of a tool condition model that produces a distribution over the current state of the machine is relatively novel, as most existing approaches just aim to predict a single value. The confidence bounds would likely prove useful in a practical application where the tool condition predictions are used to determine when to change machine tools.

The use of Welch's method to calculate the power spectrum results in a noticeably smoother spectrum, with much less noise between the frequency peaks. Welch's method also provides a way to reduce the time series from an arbitrary length to specific number of points. The transformation from the time domain to the frequency domain results in a significant dimensional reduction. The acceleration time series signal for each cutting operation contains on the order of 10,000 points and the audio time series contains on the order of 100,000 points. By applying Welch's method, these signals are each reduced to 256 discrete points in the frequency domain. This corresponds to an order of magnitude dimension reduction in the acceleration data, and two orders of magnitude reduction in the acoustic data. Whilst the loss of frequency resolution is often undesirable in signal processing applications, it provides a convenient way to reduce the dimensionality of the data in this study.

Future work could investigate how the performance of the models with SE or ARD SE kernels can be improved. The model with the SE kernel did not fit the dataset well, as a single length scale is used for all dimensions in the feature vector. It is possible that the performance of the SE kernel model could be improved by applying a normalization technique to the feature vectors. The ARD kernel over-fitted the training dataset leading to bad performance on the testing set. The overfitting problem could be solved in several ways, such as adjusting the hyperparameter optimization strategy, using cross-validation, or applying a prior to the hyperparameters [21]. Future work could investigate how a model with an ARD SE kernel could be used to automatically determine the relevance of different audio and acceleration frequencies.

In this work, we introduce a composite kernel by combining a number of kernels, each corresponds to a particular data type. The introduction of the composite kernel allows the number of model hyperparameters to be reduced and improves model generalization for this application. It would be interesting to develop a standardized representation for GPR models with composite kernels. The current PMML standard could be extended to allow covariance kernel functions to be added or multiplied.

For this method to have practical applications it must generalize well to a range of different machines and machine operations. In this work, we trained separate models for climb-cutting and conventional-cutting operations. It could be beneficial to train a single model to predict tool condition with a range of operations, given the input signal and the operation type being performed. Another interesting research direction involves using a real-time adaptive GPR learning algorithm. In this case, the GPR algorithm only retains a subset of the observed data points. Using this technique would reduce computational demands and memory requirements, while improving generalizability across different parameter spaces [29].

Acknowledgements

The authors acknowledge the support by the Smart Manufacturing Systems Design and Analysis Program at the National Institute of Standards and Technology (NIST), US Department of Commerce. This work was performed under the financial assistance award (NIST Cooperative Agreement 70NANB17H031) to Stanford University. The authors would like to acknowledge the collaboration with the late Professor David Dornfeld of University of California Berkeley, who was instrumental in making the experiments possible.

Disclaimer

Certain commercial systems are identified in this paper. Such identification does not imply recommendation or endorsement by NIST; nor does it imply that the products identified are necessarily the best available for the purpose. Further, any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NIST or any other supporting U.S. government or corporate organizations.

References

[1] M. Hermann, T. Pentek, and B. Otto, "Design Principles for Industrie 4.0 Scenarios," in *49th Hawaii International Conference on Hawaii International Conference on System Sciences (HICSS)*, 2016, pp. 3928–3937.

[2] Y. Altintas, *Manufacturing Automation: Metal Cutting Mechanics, Machine Tool Vibrations, and CNC Design*. Cambridge University Press, 2012.

[3] J. Fan, F. Han, and H. Liu, "Challenges of Big Data Analysis," *National Science Review*, vol. 1, no. 2, pp. 293–314, 2014.

[4] E. Kannatey-Asibu and D. A. Dornfeld, "A Study of Tool Wear Using Statistical Analysis of Metal-cutting Acoustic Emission," *Wear*, vol. 76, no. 2, pp. 247–261, 1982.

[5] R. G. Silva, R. L. Reuben, K. J. Baker, and S. J. Wilcox, "Tool Wear Monitoring of Turning Operations by Neural Network and Expert System Classification of a Feature Set Generated from Multiple Sensors," *Mechanical Systems and Signal Processing*, vol. 12, no. 2, pp. 319–332, 1998.

[6] A. E. Diniz, J. J. Liu, and D. A. Dornfeld, "Correlating Tool Life, Tool Wear and Surface Roughness by Monitoring Acoustic Emission in Finish Turning," *Wear*, vol. 152, no. 2, pp. 395–407, 1992.

[7] D. E. Dimla, "Sensor Signals for Tool-wear Monitoring in Metal Cutting Operations—a Review of Methods," *International Journal of Machine Tools and Manufacture*, vol. 40, no. 8, pp. 1073–1098, 2000.

[8] D. E. Dimla, P. M. Lister, and N. J. Leighton, "Neural Network Solutions to the Tool Condition Monitoring Problem in Metal Cutting—a Critical Review of Methods," *International Journal of Machine Tools and Manufacture*, vol. 37, no. 9, pp. 1219–1241, 1997.

[9] W. Sobel, "MTConnect Standard. Part 1 — Overview and Protocol," Version 1.3.0, Jun. 2015.

[10] R. Bhinge, J. Park, K. H. Law, D. A. Dornfeld, M. Helu, and S. Rachuri, "Toward a Generalized Energy Prediction Model for Machine Tools," *Journal of Manufacturing Science and Engineering*, vol. 139, no. 4, p. 041013, 2017.

[11] R. Bhinge *et al.*, "An Intelligent Machine Monitoring System for Energy Prediction Using a Gaussian Process Regression," in *2014 IEEE International Conference on Big Data*, 2014, pp. 978–986.

[12] D. E. Dimla and P. M. Lister, "On-line Metal Cutting Tool Condition Monitoring. I: Force and Vibration Analyses," *International Journal of Machine Tools and Manufacture*, vol. 40, no. 5, pp. 739–768, 2000.

[13] A. Guazzelli, M. Zeller, W. Lin, and G. Williams, "PMML: An Open Standard for Sharing Models," *The R Journal*, vol. 1, no. 1, pp. 60–65, 2009.

[14] P. Stoica and R. L. Moses, *Spectral Analysis of Signals*, vol. 452. Pearson Prentice Hall Upper Saddle River, NJ, 2005.

[15] R. L. Allen and D. Mills, *Signal Analysis: Time, Frequency, Scale, and Structure*. John Wiley & Sons, 2004.

[16] P. Welch, "The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging over Short, Modified Periodograms," *IEEE Transactions on Audio and Electroacoustics*, vol. 15, no. 2, pp. 70–73, Jun. 1967.

[17] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*. Wiley India Pvt. Limited, 2009.

[18] J. O. Smith III, *Spectral Audio Signal Processing*. W3K Publishing, 2011.

[19] C. K. Williams and C. E. Rasmussen, "Gaussian Processes for Regression," *Advances in Neural Information Processing Systems*, pp. 514–520, 1996.

[20] A. G. Wilson and R. P. Adams, "Gaussian Process Kernels for Pattern Discovery and Extrapolation.," in *ICML (3)*, 2013, pp. 1067–1075.

[21] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning*. 2006.

[22] M. Ferguson, K. H. Law, R. Bhinge, D. Dornfeld, J. Park, and Y. T. Lee, "Evaluation of a PMML-based GPR Scoring Engine on a Cloud Platform and Microcomputer Board for Smart Manufacturing," in *2016 IEEE International Conference on Big Data*, 2016, pp. 2014–2023.

[23] D. Duvenaud, "Automatic Model Construction with Gaussian Processes," University of Cambridge, 2014.

[24] C. E. Rasmussen and H. Nickisch, "Documentation for GPML Matlab Code, Version 4.1," 27-Nov-2017. [Online]. Available: http://www.gaussianprocess.org/gpml/code/matlab/doc/. [Accessed: 18-Jan-2018].

[25] H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis*, vol. 7. Cambridge University Press, 2004.

[26] M. Zeller *et al.*, "Open Standards and Cloud Computing: KDD-2009 Panel Report," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 11–18.

[27] The Data Mining Group, "The Predictive Model Markup Language, Version 4.3." 01-Aug-2016.

[28] J. Park *et al.*, "Gaussian Process Regression (GPR) Representation Using Predictive Model Markup Language (PMML)," *Smart and Sustainable Manufacturing Systems. Manuscript submitted for publication*, 2017.

[29] J. Quiñonero-Candela and C. E. Rasmussen, "A Unifying View of Sparse Approximate Gaussian Process Regression," *Journal of Machine Learning Research*, vol. 6, no. Dec, pp. 1939–1959, 2005.

Figure Captions List

Fig. 1	Experimental setup in a Mori Seiki NVD1500DCG milling machine showing (1) the cutting tool and (2) the sensor unit from Infinite Uptime
Fig. 2	Solid carbide end mill flute in different states of condition. A lower value of y indicates higher tool wear condition
Fig. 3	Flow diagram showing the proposed data processing pipeline. In the pipeline, raw time series data from the milling machine sensors is mapped to predicted tool condition values.
Fig. 4	Acceleration time series measured while a single part was being produced. The time series data shows that the machine performs a series of repetitive actions, separated by brief idle periods.
Fig. 5	The acceleration time series after each action was labelled using data from the machine controller
Fig. 6	Comparison of the acceleration periodograms recorded with a sharp tool and a worn tool, while a climb-cutting operation was being performed
Fig. 7	Comparison of the audio periodograms recorded with a sharp tool and a worn tool, while a climb-cutting operation was being performed
Fig. 8	Tool condition prediction for three testing data sets using the model with the SSE kernel. Each plot represents a test where the milling machine was run until the cutting tool became severely worn or broken. The shaded region represents the 90% confidence interval for each prediction.
Fig. 9	Tool condition prediction for three testing data sets using the model with the SSE kernel. For these plots, the previous state c^i was omitted from the feature vector when training and testing the model. The shaded region represents the 90% confidence interval for each prediction.
Fig. 10	Predictive model markup language (PMML) representation of the Gaussian process regression (GPR) model use to predict the condition of machine tools when performing climb-cutting operations. In this model the squared exponential (SE) covariance kernel is used.

Table Caption List

Table 1	Experimental statistics and operating parameters for the Mori Seiki NVD1500DCG milling machine
Table 2	Model performance on training set and testing set. Performance is measured using root mean squared error (RMSE). A smaller value of RMSE indicates a better fit.

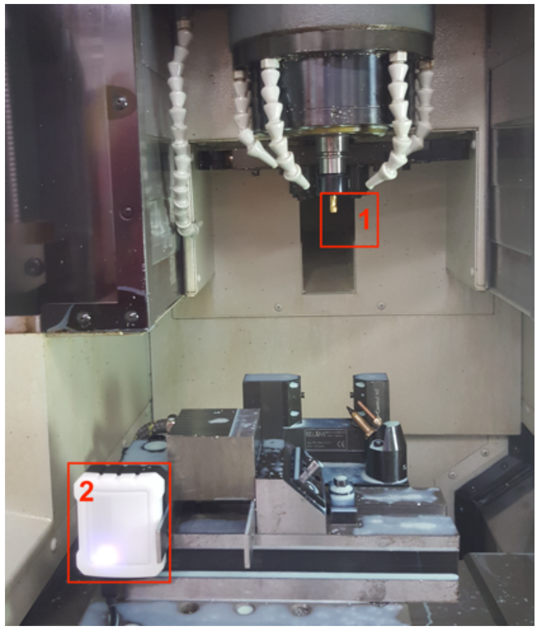


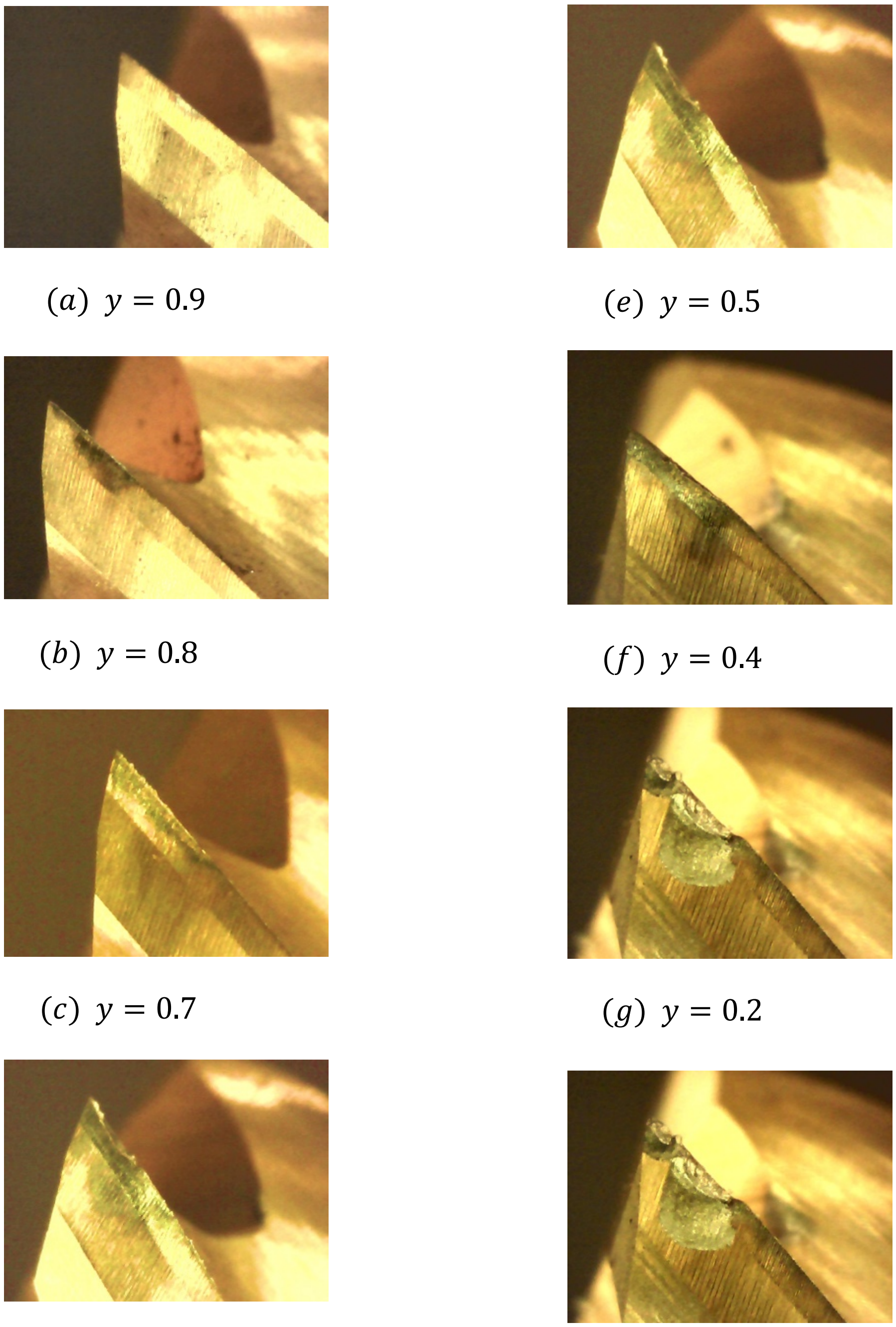
Figure 1. Experimental setup in a Mori Seiki NVD1500DCG milling machine showing (1) the cutting tool and (2) the sensor unit from Infinite Uptime

Table 1. Experimental statistics and operating parameters for the Mori Seiki NVD1500DCG milling machine

Parameter/Statistic	Value
Parts produced	79
Machine tools used	18
Machine feed rate	304.4 mm/minute
Machine tool rotation rate	3000 revolutions per minute (RPM)

Table 2. Model performance on training set and testing set. Performance is measured using root mean squared error (RMSE). A smaller value of RMSE indicates a better fit.

Model	**RMSE**	
	Training Set	Testing Set
GP with squared exponential (SE) kernel	0.102	0.151
GP with ARD squared exponential (ARD SE) kernel	**0.011**	0.191
GP with sum of squared exponential (SSE) kernel	0.021	**0.063**



$(a)\ y = 0.9$

$(b)\ y = 0.8$

$(c)\ y = 0.7$

$(d)\ y = 0.6$

$(e)\ y = 0.5$

$(f)\ y = 0.4$

$(g)\ y = 0.2$

$(h)\ y = 0.1$

Figure 2. Solid carbide end mill flute in different states of condition. A lower value of y indicates higher tool wear condition

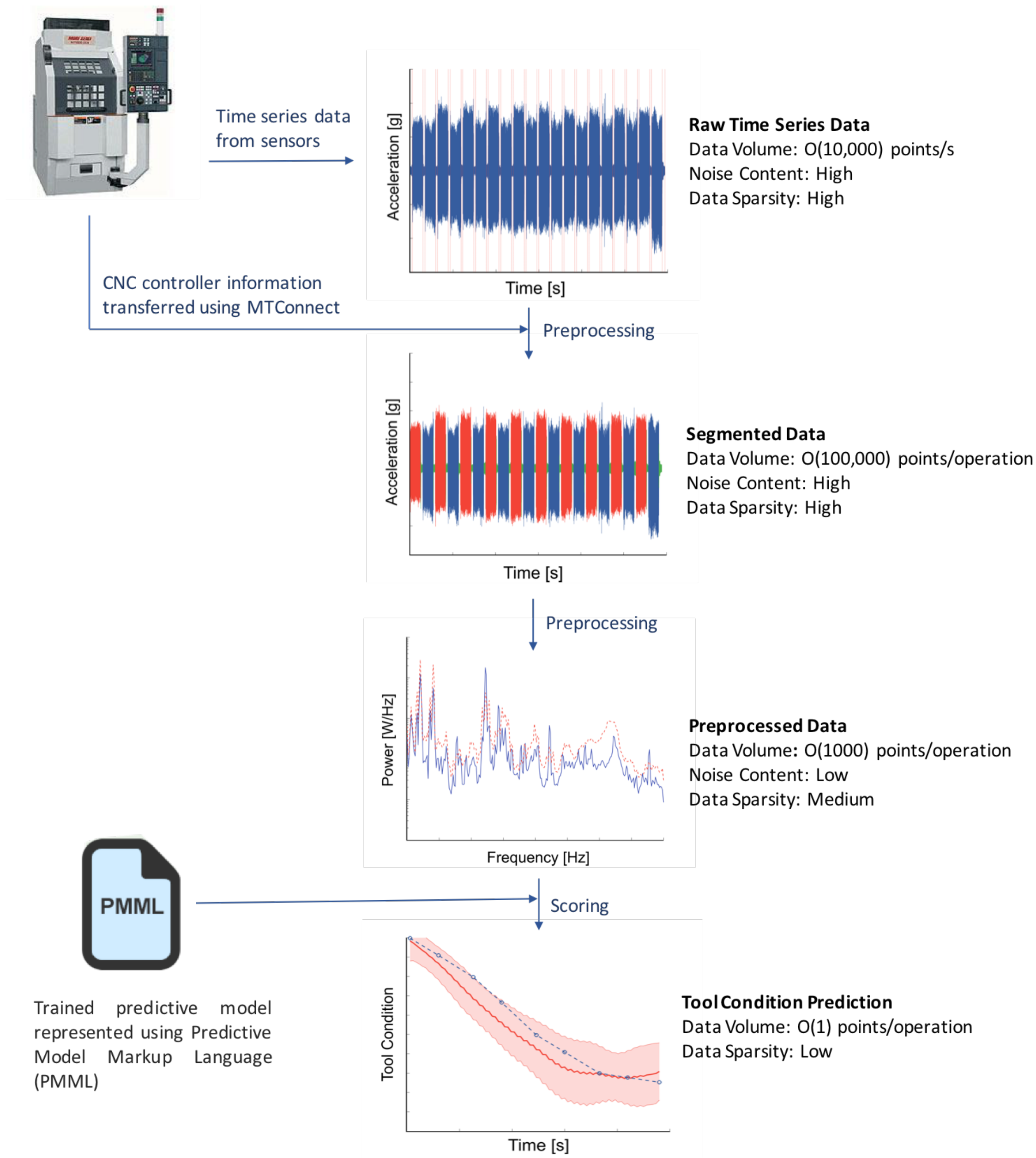


Figure 3. Diagram showing the proposed data processing pipeline. In the pipeline, raw time series data from the milling machine sensors is mapped to tool condition predictions.

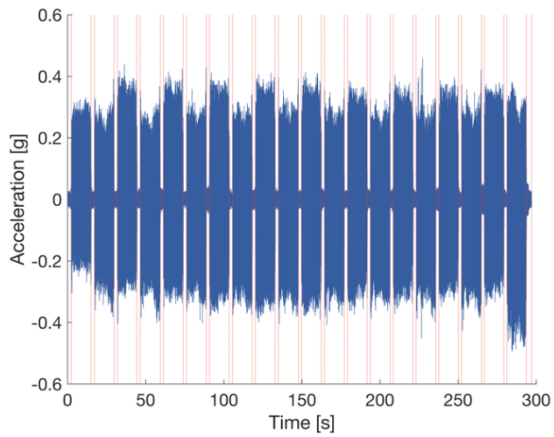


Figure 4. Acceleration time series measured while a single part was being produced. The time series data shows that the machine performs a series of repetitive actions, separated by brief idle periods.

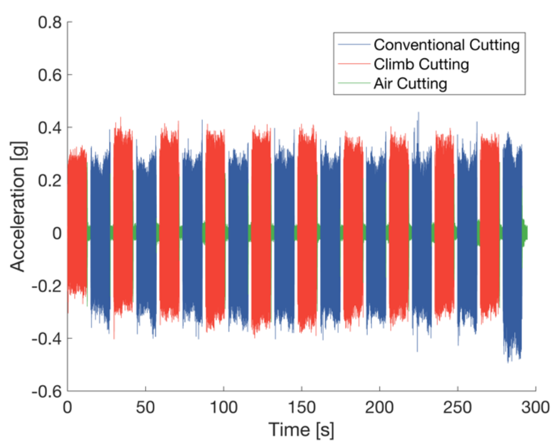


Figure 5. The acceleration time series after each action was labelled using data from the machine controller

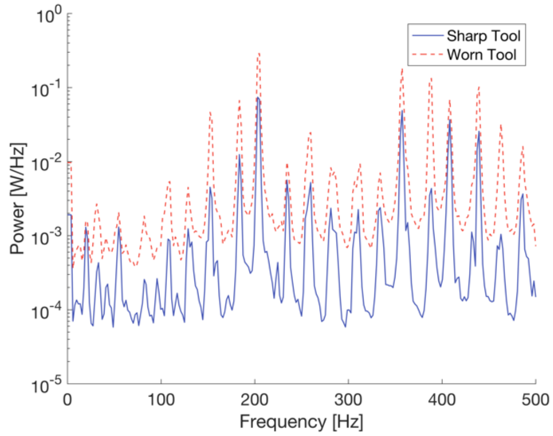


Figure 6. Comparison of the acceleration periodograms recorded with a sharp tool and a worn tool, while a climb-cutting operation was being performed

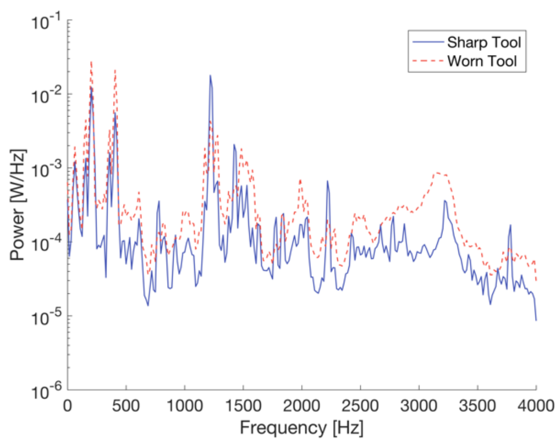


Figure 7. Comparison of the audio periodograms recorded with a sharp tool and a worn tool, while a climb-cutting operation was being performed

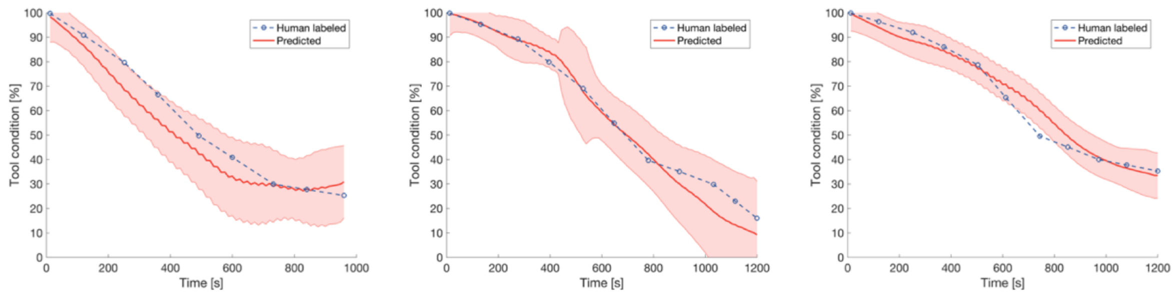


Figure 8. Tool condition prediction for three testing data sets using the model with the SSE kernel. Each plot represents a test where the milling machine was run until the cutting tool became severely worn or broken. The shaded region represents the 90% confidence interval for each prediction.

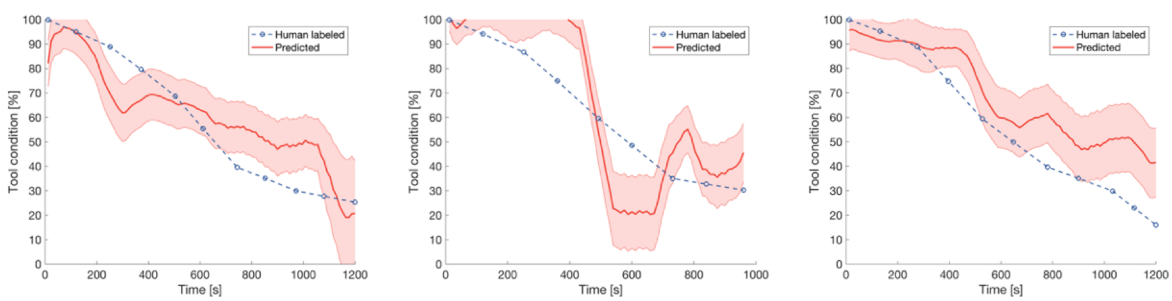


Figure 9. Tool condition prediction for three testing data sets using the model with the SSE kernel. For these plots, the previous state c^i was omitted from the feature vector when training and testing the model. The shaded region represents the 90% confidence interval for each prediction.

```xml
<?xml version="1.0" encoding="utf-8"?>
<pmml xmlns="http://www.dmg.org/PMML-4_3" version="4.3">
  <header copyright="DMG.org" />
  <datadictionary numberoffields="516">
  <datafield datatype="double" name="x1" optype="continuous" />
  ...
  <datafield datatype="double" name="x515" optype="continuous" />
  <datafield datatype="double" name="y1" optype="continuous" />
  </datadictionary>
  <gaussianprocessmodel functionname="regression" modelname="Gaussian Process Model">
    <miningschema>
    <miningfield name="x1" usagetype="active" />
    ...
    <miningfield name="x515" usagetype="active" />
    <miningfield name="y1" usagetype="predicted" />
    </miningschema>
    <output>
      <outputfield datatype="double" feature="predictedValue" name="MeanValue" optype="continuous" />
      <outputfield datatype="double" feature="predictedValue" name="StandardDeviation" optype="continuous" />
    </output>
    <squaredexponentialkernel gamma="0.099378" noisevariance="0.00056713">
      <lambda>
        <array n="1" type="real">0.114819</array>
      </lambda>
    </squaredexponentialkernel>
    <traininginstances fieldcount="516" istransformed="false" recordcount="435">
      <instancefields>
      <instancefield column="x1" field="x1" />
      ...
      <instancefield column="x515" field="x515" />
      <instancefield column="y1" field="y1" />
      </instancefields>
      <inlinetable>
        <row>
          <x1>0.0016105</x1
          ...
          <x515>1.00000</x51>
          <y1>0.16998</y1>
        </row>
        ...
      </inlinetable>
    </traininginstances>
  </gaussianprocessmodel>
</pmml>
```

Figure 10. Predictive Model Markup Language (PMML) representation of the Gaussian Process Regression (GPR) model use to predict the condition of machine tools when performing climb-cutting operations. In this model, the squared exponential (SE) covariance kernel is used.