Proceedings of the ASME 2017 12th International Manufacturing Science and Engineering Conference
MSEC2017
June 4-8, 2017, Los Angeles, CA, USA

MSEC2017-2679

Data-Driven Prognostics Using Random Forests: Prediction of Tool Wear

Dazhong Wu[1], Connor Jennings[1], Janis Terpenny[1], Robert Gao[2], Soundar Kumara[1]

[1]Department of Industrial and Manufacturing Engineering, Pennsylvania State University, University Park PA 16802, USA
[2]Department of Mechanical and Aerospace Engineering, Case Western Reserve University, Cleveland OH 44106, USA

ABSTRACT

Manufacturers have faced an increasing need for the development of predictive models that help predict mechanical failures and remaining useful life of a manufacturing system or its system components. Model-based or physics-based prognostics develops mathematical models based on physical laws or probability distributions, while an in-depth physical understanding of system behaviors is required. In practice, however, some of the distributional assumptions do not hold true. To overcome the limitations of model-based prognostics, data-driven methods have been increasingly applied to machinery prognostics and maintenance management, transforming legacy manufacturing systems into smart manufacturing systems with artificial intelligence. While earlier work demonstrated the effectiveness of data-driven approaches, most of these methods applied to prognostics and health management (PHM) in manufacturing are based on artificial neural networks (ANNs) and support vector regression (SVR). With the rapid advancement in artificial intelligence, various machine learning algorithms have been developed and widely applied in many engineering fields. The objective of this research is to explore the ability of random forests (RFs) to predict tool wear in milling operations. The performance of ANNs, SVR, and RFs are compared using an experimental dataset. The experimental results have shown that RFs can generate more accurate predictions than ANNs and SVR in this experiment.

Keywords: Tool wear prediction; Predictive modeling; Machine learning; Random forests (RFs); Support vector machines (SVMs); Artificial neural networks (ANNs); Prognostics and health management

INTRODUCTION

Smart manufacturing aims to integrate big data, advanced analytics, high performance computing, and Industrial Internet of Things (IIoT) into traditional manufacturing systems and processes to create highly customizable products with higher quality at lower costs. As opposed to traditional factories, a smart factory utilizes interoperable information and communications technologies (ICT), intelligent automation systems, and sensor networks to monitor machinery conditions, diagnose the root cause of failures, and predict the remaining useful life (RUL) of mechanical systems or components. For example, almost all engineering systems (e.g., aerospace systems, nuclear power plants, and machine tools) are subject to mechanical failures resulting from deterioration with usage and age or abnormal operating conditions [1-3]. Some of the typical failure modes include excessive temperature, stress, deflection, fracture, fatigue, corrosion, and wear. The degradation and failures of engineering systems or system components will often incur higher costs and lower productivity due to unexpected machine downtime. In order to increase manufacturing productivity while reducing maintenance costs, it is crucial to develop and implement an intelligent maintenance strategy that allows manufacturers to determine the conditions of in-service systems or components in order to predict when maintenance should be performed.

Conventional maintenance strategies include reactive, preventive, and proactive maintenance [4-6]. The most basic approach to maintenance is reactive, also known as run-to-failure maintenance planning. In the reactive maintenance strategy, assets are deliberately allowed to operate until failures actually occur. The assets are maintained on an as-needed basis. One of the disadvantages of reactive maintenance is that it is difficult to anticipate the maintenance resources (e.g., manpower, tools and replacement parts) that will be required for repairs. Preventive maintenance is often referred to as use-based maintenance. In preventive maintenance, maintenance activities are performed after a predetermined period of time based on the estimated probability that the systems or components will fail in the specified time interval. Although preventive maintenance allows for more consistent and predictable maintenance schedules, more maintenance activities are often needed as opposed to reactive maintenance. To improve the efficiency and effectiveness of preventive maintenance, predictive maintenance is an alternative strategy in which maintenance actions are scheduled based on equipment performance or conditions instead of time or usage. The objective of proactive maintenance is to determine the conditions of in-service equipment, and ultimately to predict the time at which a system or a component will no longer perform its intended function.

The discipline that predicts health condition and remaining useful life (RUL) based on previous and current operating conditions is often referred to as prognostics and health management (PHM). Prognostic approaches fall into two categories: model-based and data-driven prognostics [7-12]. Model-based prognostics refers to approaches based on mathematical models of system behavior derived from physical laws or probability distributions. For example, model-based prognostics include methods based on Wiener and Gamma processes [13], hidden Markov models (HMMs) [14], Kalman filters [15], and particle filters [16, 17]. Model-based prognostics is often used in applications where mathematical representations can be developed to describe physical phenomena or failure modes. The main advantage of model-based approaches is the ability to incorporate physical understanding of system behaviors. However, one of the limitations of model-based prognostics is that an in-depth understanding of underlying physics that leads to system failures is required. Such knowledge is sometimes not readily available for complex systems.

In comparison with model-based prognostics, data-driven prognostics refers to approaches that build a predictive model using a learning algorithm. For example, classical data-driven prognostics include approaches based on artificial neural networks (ANNs), k-means clustering, logistic regression, and support vector machines (SVMs). While data-driven approaches may require large volumes of quality training data, it allows one to gain in-depth insights into complex phenomena that are not well understood. In addition, data-driven approaches sometimes make more accurate predictions than physics-based models because mathematical models cannot accommodate sufficient complexity.

Over the past decade, a few classical machine learning algorithms such as ANNs and SVMs have been applied in the area of data-driven prognostics. However, little research has been conducted to explore the potential of new machine learning algorithms and compare their performance with ANNs and SVMs [18]. Because RFs have the potential to handle a large number of input variables without variable selection and they do not overfit, we investigate the ability of RFs for the prediction of tool wear using an experimental dataset. Further, the performance of RFs is compared with that of ANNs and SVMs using accuracy and training time.

The main contributions of this paper include:

- Tool wear in milling operations is predicted using RFs along with cutting force, vibration, and acoustic emission signals. Experimental results have shown that the predictive model trained by RFs is very accurate. To the best of our knowledge, the random forest algorithm is applied to predict tool wear in the manufacturing domain for the first time.
- The performances of ANNs, SVMs, and RFs are compared using an experimental dataset with respect to the accuracy of regression (e.g., mean squared error and coefficient of determination) and training time. While the training time for RFs is longer than that of ANNs and SVMs, the predictive model built by RFs is the most accurate for the application example.

The remainder of the paper is organized as follows: Section 2 reviews the related literature on data-driven methods for tool wear prediction. Section 3 presents the methodology for tool wear prediction using ANNs, SVMs, and RFs. Section 4 presents an experimental setup and the experimental dataset acquired from different types of sensors (e.g., cutting force sensor, vibration sensor, acoustic emission sensor) on a CNC milling machine. Section 5 presents experimental results, demonstrates the effectiveness of the three machine learning algorithms, and compares the performance of each. Section 6 provides conclusions that include a discussion of research contribution and future work.

2. Data-Driven Methods for Tool Wear Prediction

Tool wear is the most commonly observed and unavoidable phenomenon in manufacturing processes such as drilling, milling, and turning [19-21]. The rate of tool wear is typically affected by process parameters (e.g., cutting speed and feed rate), cutting tool geometry, and properties of workpiece and tool materials. Taylor's equation for tool life expectancy [22] provides an approximation of tool wear. However, with the rapid advancement of sensing technology and increasing number of sensors equipped on modern CNC machines, it is possible to predict tool wear more accurately using various measurement data. This section presents a review of data-driven methods for tool wear prediction.

Schwabacher and Goebel [23] conducted a review of data-driven methods for prognostics. The most popular data-driven approaches to prognostics include ANNs, decision trees, and SVMs in the context of systems health management. ANNs are a family of computational models based on biological neural networks which are used to estimate complex relationships between inputs and outputs. Bukkapatnam *et al.* [24-26] developed effective tool wear monitoring techniques using ANNs based on features extracted from the principles of nonlinear dynamics. Özel and Karpat [27] presented a predictive modeling approach for surface roughness and tool wear for hard turning processes using ANNs. The inputs of the ANN model include workpiece hardness, cutting speed, feed rate, axial cutting length, and mean values of three force components. Experimental results have shown that the model trained by ANNs provides accurate predictions of surface roughness and tool flank wear. Palanisamy *et al.* [28] developed a predictive model for predicting tool flank wear in end milling operations using feed-forward back propagation (FFBP) ANNs. Experimental results have shown that the predictive model based on ANNs can make accurate predictions of tool flank wear using cutting speeds, feed rates, and depth of cut. Sanjay *et al.* [29] developed a model for predicting tool flank wear in drilling using ANNs. The feed rates, spindle speeds, torques, machining times, and thrust forces are used to train the ANN model. The experimental results have demonstrated that ANNs can predict tool wear accurately. Chungchoo and Saini [30] developed an online fuzzy neural network (FNN) algorithm that estimates the average width of flank wear and maximum depth of crater wear. A modified least-square backpropagation neural network was built to estimate flank and crater wear based on cutting force and acoustic emission signals. Chen and Chen [31] developed an in-process tool wear prediction system using ANNs for milling operations. A total of 100 experimental data were used for training the ANN model. The input variables include feed rate, depth of cut, and average peak cutting forces. The ANN model can predict tool wear with an error of 0.037mm on average.

Cho *et al.* [32] developed an intelligent tool breakage detection system with the SVM algorithm by monitoring cutting forces and power consumption in end milling processes. Linear and polynomial kernel functions were applied in the SVM algorithm. It has been demonstrated that the predictive model built by SVMs can recognize process abnormalities in milling. Benkedjouh *et al.* [33] presented a method for tool wear assessment and remaining useful life prediction using SVMs. The features were extracted from cutting force, vibration, and acoustic emission signals. The experimental results have shown that SVMs can be used to estimate the wear progression and predict RUL of cutting tools effectively.

Another data-driven method for prognostics is based on decision trees. Decision trees are a non-parametric supervised learning method used for classification and regression. The goal of decision tree learning is to create a model that predicts the value of a target variable by learning decision rules inferred from data features. A decision tree is a flowchart-like structure in which each internal node denotes a test on an attribute, each branch represents the outcome of a test, and each leaf node holds a class label. Jiaa and Dornfeld [34] proposed a decision tree-based method for the prediction of tool flank wear in a turning operation using acoustic emission and cutting force signals. The features characterizing the AE RMS and cutting force signals were extracted from both time and frequency domains. The decision tree approach was demonstrated to be able to make reliable inferences and decisions on tool wear classification. Elangovan *et al.* [35] developed a decision tree-based algorithm for tool wear prediction using vibration signals. Ten-fold cross-validation was used to evaluate the accuracy of the predictive model created by the decision tree algorithm. The maximum classification accuracy was 87.5%.

In summary, the related work presented in this section builds on previous research to explore how the conditions of tool wear can be monitored as well as how tool wear can be predicted using predictive modeling. While earlier work focused on the prediction of tool wear using ANNs, SVMs, and decision trees, this paper explores the potential of a new method, random forests, for tool wear prediction. Further, the performance of RFs is compared with that of ANNs and SVMs. Because RFs are an extension of decision trees, the performance of RFs is not compared with that of decision trees.

3. Methodology

This section presents the methodology for data-driven prognostics for tool wear prediction using ANNs, SVR, and RFs. The input of ANNs, SVR, and RFs is the following labeled training data: $D = (x_i, y_i)$ where $x_i = (\mathrm{F}_X, \mathrm{F}_Y, \mathrm{F}_Z, \mathrm{V}_X, \mathrm{V}_Y, \mathrm{V}_Z, AE), y_i \in \mathbb{R}$. The description of these input data can be found in Table 1.

Table 1. Signal Channel and Data Description

Signal Channel	Data Description
Channel 1	F_X: Force (N) in X dimension
Channel 2	F_Y: Force (N) in Y dimension
Channel 3	F_Z: Force (N) in Z dimension
Channel 4	V_X: Vibration (g) in X dimension
Channel 5	V_Y: Vibration (g) in Y dimension
Channel 6	V_Z: Vibration (g) in Z dimension
Channel 7	AE: Acoustic Emission (V)

3.1 Tool Wear Prediction Using ANNs

The original ANN algorithm was introduced by McCulloch and Pitts [36]. ANNs are a family of models inspired by biological neural networks. An ANN is defined by three types of parameters: (1) the interconnection pattern between different layers of neurons, (2) the learning process for updating the weights of the interconnections, and (3) the activation function that converts a neuron's weighted input to its output activation. Among many types of ANNs, the feed-forward neural network is the first and the most popular ANN. In a feed-forward ANN, connections between units do not form a cycle or loop. Back-propagation is a learning algorithm for training ANNs in conjunction with an optimization method such as gradient descent. Back-propagation is used to calculate the gradient of a loss function with respect to all the weights in a ANN. The gradient is fed to the optimization method which in turn uses it to update the initial weights for minimizing the loss/error function.

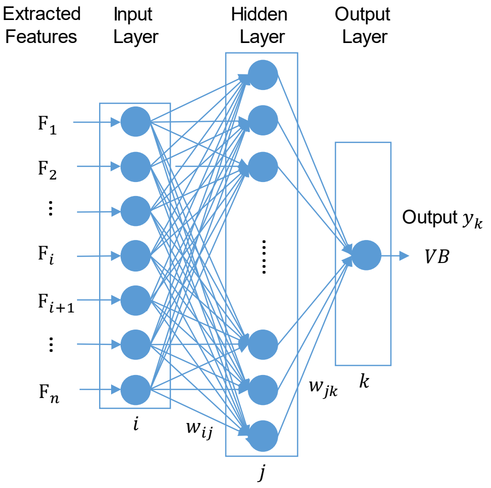

Figure 1 Tool wear prediction using a ANN

Figure 1 illustrates the architecture of the feed-forward back-propagation (FFBP) ANN with a single hidden layer. In the context of tool wear prediction, the ANN has three layers, including input layer i, hidden layer j, and output layer k. Each layer consists of one or more neurons or units, represented by the circles. The flow of information is represented by the lines between the units. The first layer has input neurons which act as buffers for distributing the extracted features (i.e., F_i) from the input data (i.e., x_i). The number of the neurons in the input layer is the same as that of extracted features from input variables. Each value from the input layer is duplicated and sent to all neurons in the hidden layer. The hidden layer is used to process and connect the information from the input layer to the output layer in a forward direction. Specifically, these values entering a neuron in the hidden layer are multiplied by weights w_{ij}. Initial weights are randomly selected between 0 and 1. A neuron in the hidden layer sums up the weighted inputs and generates a single value. This value is the input of an activation function in the hidden layer f_h (e.g., sigmoid function) that converts the weighted input to the output of the neuron. Similarly, the outputs of all the neurons in the hidden layer are multiplied by weights w_{jk}. A neuron in the output layer sums up the weighted inputs and generates a single value. An activation function in the output layer f_o converts the weighted input to the predicted output y_k of the ANN, which is the predicted flank wear VB. The output layer has only one neuron because there is only one response variable.

The performance of ANNs depends on the topology or architecture of ANNs (i.e., the number of layers) and the number of neurons in each layer. However, there are no standard or well-accepted methods or rules for determining the number of hidden layers and neurons in each hidden layer. According to Heaton [37] and Blum [38], there are some empirically-derived rules. For example, Rule #1: an ANN with a single hidden layer is sufficient for the majority of problems; Rule #2: the optimal size of a hidden layer is usually between the size of input and output layers; Rule #3: if the data is linearly separable, then hidden layers are not needed. It should be noted that the above rules are not always true because the number of neurons also depends on the complexity of activation functions, training algorithms, and training datasets. In this research, the single-hidden-layer ANN with 16 neurons in the hidden layer is selected. The termination criterion of the training algorithm is that training stops if the fit criterion (i.e., least squares) falls below 1.0e-4.

3.2 Tool Wear Prediction Using SVR

The original SVM for regression was developed by Vapnik *et al.* [39, 40]. A comprehensive tutorial on SVM can be found in Smola and Schölkopf [41]. A SVM constructs a hyperplane or set of hyperplanes in a high or infinite-dimensional space, which can be used for classification and regression.

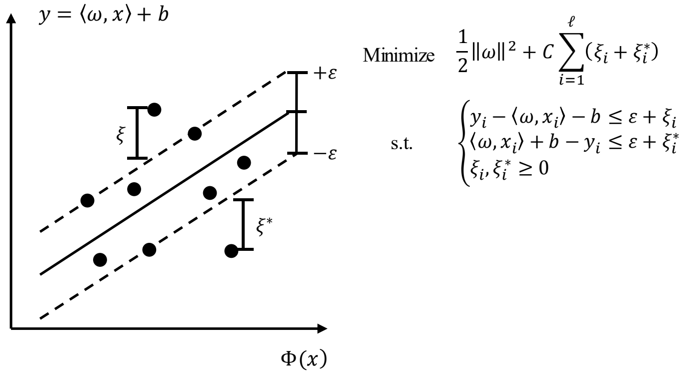
$y = \langle \omega, x \rangle + b$
$+\varepsilon$
$-\varepsilon$
ξ
ξ^*
$\Phi(x)$
Minimize $\frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{\ell}(\xi_i + \xi_i^*)$
s.t. $\begin{cases} y_i - \langle \omega, x_i \rangle - b \le \varepsilon + \xi_i \\ \langle \omega, x_i \rangle + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases}$

Figure 2 Tool wear prediction using SVR

Figure 2 illustrates the framework of predicting tool wear using SVR for linear cases. In the context of tool wear prediction, the objective is to minimize the error expressed by Equation (3.1)

$$\begin{aligned} &\text{Minimize} && \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{\ell}(\xi_i + \xi_i^*) \\ &\text{Subject to} && \begin{cases} y_i - \langle \omega, x_i \rangle - b \le \varepsilon + \xi_i \\ \langle \omega, x_i \rangle + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases} \end{aligned} \tag{3.1}$$

where $\omega \in \chi, C = 1$, $\varepsilon = 0.1$, and $\xi_i, \xi_i^* = 0.001$. b can be computed as follows:

$$\begin{aligned} b = y_i - \langle \omega, x_i \rangle - \varepsilon \quad \text{for } \alpha_i \in [0, C] \\ b = y_i - \langle \omega, x_i \rangle + \varepsilon \quad \text{for } \alpha_i^* \in [0, C] \end{aligned} \tag{3.2}$$

For non-linear SVR, the training patterns x_i can be preprocessed by a non-linear kernel function $k(x, x') := \langle \Phi(x), \Phi(x)' \rangle$ where $\Phi(x)$ is a transformation that maps x to a high-dimensional space. These kernel functions need to satisfy the Mercer's theorem. Many kernels have been developed for various applications. The most popular kernels include linear, polynomial, Gaussian radial basis function (RBF), and sigmoid. In many applications, a nonlinear kernel function provides better accuracy. According to the literature [32, 33], the Gaussian RBF kernel is one of the most effective kernel functions used in tool wear prediction. In this research, the Gaussian RBF kernel is used to transform the input dataset $D = (x_i, y_i)$ where x_i is the input vector and y_i is the response variable (i.e., flank wear) into a new dataset in a high-dimensional space. The new dataset is linearly separable by a hyperplane in a higher-dimensional Euclidean space as illustrated in Figure 2. The points outside the two dashed lines are called support vectors. The slack variables ξ_i and ξ_i^* are introduced in the instances where the constraints are infeasible. The slack variables denote the deviation from predicted values with the error of $\varepsilon = 0.1$. The RBF kernel is $k(x_i, x_j) = exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$ where $\sigma^2 = 0.5$. At the optimal solution, we obtain

$$\omega = \sum_{i=1}^{\ell}(\alpha_i - \alpha_i^*)\Phi(x) \text{ and } f(x) = \sum_{i=1}^{\ell}(\alpha_i - \alpha_i^*)k(x_i, x_j) + b \tag{3.3}$$

3.3 Tool Wear Prediction Using RFs

The random forest algorithm, developed by Leo Breiman [42, 43], is an ensemble learning method that constructs a forest of decision trees from bootstrap samples of a training dataset. Each decision tree produces a response, given a set of predictor values. In a decision tree, each internal node represents a test on an attribute, each branch represents the outcome of the test, each leaf node represents a class label for classification or a response for regression. A decision tree in which the response is continuous is also referred to as a regression tree. In the context of tool wear prediction, each individual decision tree in a random forest is a regression tree because tool wear describes the gradual failure of cutting tools. A comprehensive tutorial on RFs can be found in [42-44]. Some of the important concepts related to RFs, including bootstrap aggregating or bagging, splitting and stopping criterion, are introduced in the following section.

Bootstrap aggregating or bagging: Given a training dataset D, bootstrap aggregating or bagging generates B new training datasets D_i of size N by sampling from the original training dataset D with replacement. D_i is referred to as a bootstrap sample. By sampling with replacement or bootstrapping, some observations may be repeated in each D_i. Bagging helps reduce variance and avoid overfitting. The number of regression trees B is a parameter specified by users. Typically, a few hundred to several thousand trees are used in the random forest algorithm.

Choosing variables to split on: For each of the bootstrap samples, grow an un-pruned regression tree with the following procedure: At each node, randomly sample m variables and choose the best split among those variables rather than choosing the best split among all predictors. This process is sometimes called "feature bagging". The reason why a random subset of the predictors or features is selected is because the correlation of the trees in an ordinary bootstrap sample can be reduced.

Splitting criterion: Suppose that a partition is divided into M regions R_1, R_2, …, R_m. The response is modeled as a constant c_m in each region:

$$f(x) = \sum_{m=1}^{M} c_m I(x\epsilon R_m) \tag{3.4}$$

The splitting criterion at each node is to minimize the sum of squares. Therefore, the best $\widehat{c_m}$ is the average of y_i in region R_m:

$$\widehat{c_m} = ave(y_i|x_i\epsilon R_m) \tag{3.5}$$

Consider a splitting variable j and split point s, and define the pair of half-planes

$$R_1(j,s) = \{X|X_j \leq s\} \text{ and } R_2(j,s) = \{X|X_j \geq s\}. \tag{3.6}$$

The splitting variable j and split point s should satisfy

$$\min_{j,s}\left[\min_{c1} \sum\nolimits_{x_i\epsilon R_1(j,s)} (y_i - c1)^2 + \min_{c2} \sum\nolimits_{x_i\epsilon R_2(j,s)} (y_i - c2)^2\right]. \tag{3.7}$$

For any j and s, the inner minimization is solved by

$$\widehat{c_1} = ave\big(y_i\big|x_i\epsilon R_1(j,s)\big) \text{ and } \widehat{c_2} = ave\big(y_i\big|x_i\epsilon R_2(j,s)\big). \tag{3.8}$$

Having found the best split, the dataset is partitioned into two resulting regions and repeat the splitting process on each of the two regions. This splitting process is repeated until a predefined stopping criterion is satisfied.

Stopping criterion: Tree size is a tuning parameter governing the complexity of a model. The stopping criterion is that the splitting process proceeds until the number of instances in D_i falls below a threshold. 5 is used as the threshold.

After B such trees $\{T_b\}_1^B$ are constructed, a prediction at a new point x can be made by averaging the predictions from all the individual B regression trees on x:

$$\hat{f}_{rf}^B(x) = \frac{1}{B}\sum\nolimits_{b=1}^{B} T_b(x) \tag{3.9}$$

The random forest algorithm [42, 44] for regression is as follows:

1. Draw a bootstrap sample Z of size N from the training data.
2. For each bootstrap sample, construct a regression tree by splitting a node into two children nodes until the stopping criterion is satisfied.
3. Output the ensemble of trees $\{T_b\}_1^B$.
4. Make a prediction at a new point x by aggregating the predictions of the B trees.

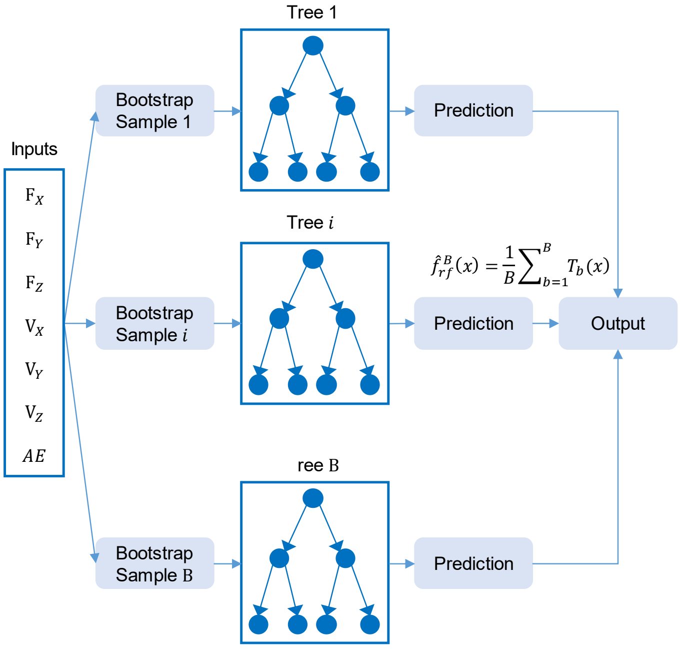

Figure 3 Tool wear estimation using a RF

Figure 3 illustrates a framework for predicting flank wear using a RF. A random forest is constructed using B= 500 regression trees. Given the labeled training dataset $D = (x_i, y_i)$, a bootstrap sample of size $N = 630$ is drawn from the training dataset. For each regression tree, $m = 9$ variables are selected at random from the 28 variables (Typically, the number of selected variables is one third of the total number of variables). The best variable/split-point is selected among the 9 variables. A regression tree progressively splits the training dataset into two child nodes, left node (with samples < z) and right node (with samples >= z). A splitting variable and split point are selected by solving Equations 3.7 and 3.8. The process is applied recursively on the dataset in each child node. The splitting process stops if the number of instances in a node is less than 5. An individual regression tree is built by starting at the root node of the tree, performing a sequence of tests about the predictors, and organizing the tests in a hierarchical binary tree structure as shown in Figure 4. After 500 trees are constructed, a prediction at a new point can be made by averaging the predictions from all the individual binary regression trees on this point.

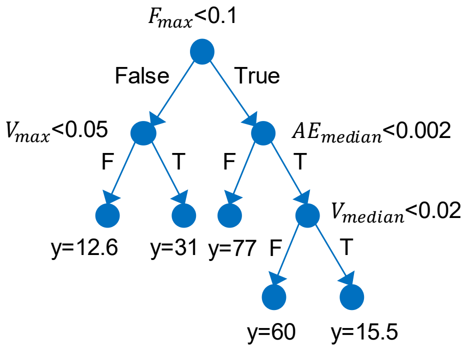

Figure 4 Binary regression tree growing process

4. Experimental Setup

The data used in this paper was obtained from *Li et al.* [45]. Some details of the experiment are presented in this section. The experimental setup is shown in Figure 5.

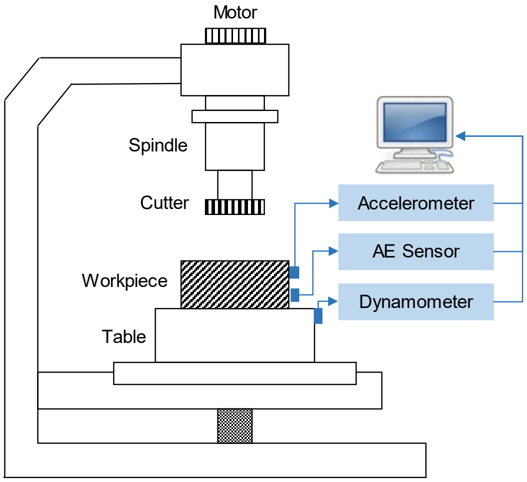

Figure 5 Experimental Setup

The experiment was conducted on a three-axis high-speed CNC machine (Röders Tech RFM 760). The workpiece material used in the dry milling experiment was stainless steel. The detailed description of the operating conditions in the dry milling operation can be found in Table 2. The spindle speed of the cutter was 10,400 RPM. The feed rate was 1,555 mm/min. The Y depth of cut (radial) was 0.125 mm. The Z depth of cut (axial) was 0.2 mm. The sampling rate was 50 KHz/channel.

Table 2. Operating Conditions

Parameter	Value
Spindle Speed	10400 RPM
Feed Rate	1555 mm/min
Y Depth of Cut	0.125 mm
Z Depth of Cut	0.2 mm
Sampling Rate	50 KHz/channel
Material	Stainless steel

As shown in Table 1, seven signal channels, including cutting force, vibration, and acoustic emission data, were monitored. A stationary dynamometer, mounted on the table of the CNC machine, was used to measure cutting forces in three, mutually perpendicular axes (x, y, and z dimensions). Three piezo accelerometers, mounted on the workpiece, were used to measure vibration in three, mutually perpendicular axes (x, y, and z dimensions). An acoustic emission (AE) sensor, mounted on the workpiece, was used to monitor a high-frequency oscillation that occurs spontaneously within metals due to crack formation or plastic deformation. Acoustic emission is caused by the release of strain energy as the microstructure of the material is rearranged. Three datasets were generated. Each dataset contains 315 individual data acquisition files in CSV format. The size of each dataset is about 2.89 GB.

5. Results and Discussions

In machine learning, feature extraction is an essential preprocessing step in which raw data collected from various signal channels is converted into a set of statistical features in a format supported by machine learning algorithms. The statistical features are then given as an input to a machine learning algorithm. In this experiment, the raw data was collected from (1) cutting force, (2) vibration, and (3) acoustic emission signal channels. A set of statistical features (28 features) extracted from these signals include Maximum, Median, Mean, and Standard Deviation as listed in Table 3.

Table 3. List of Extracted Features

Cutting Force (X, Y, Z dimensions)	Vibration (X, Y, Z dimensions)	Acoustic Emission
Max	Max	Max
Median	Median	Median
Mean	Mean	Mean
Standard Deviation	Standard Deviation	Standard Deviation

Three predictive models were developed using ANNs, SVR, and RFs, respectively. Two-thirds (2/3) of the input data (i.e., three datasets) was selected at random for model development (training). The remainder (1/3) of the input data was used for model validation (testing). Figures 6, 7, and 8 show the predicted against observed tool wear values using ANNs, SVR, and RFs, respectively.

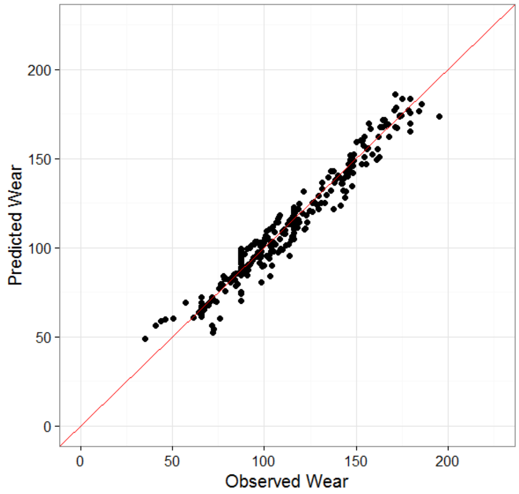

Figure 6 Comparison of observed and predicted tool wear using an ANN with 16 neurons in the hidden layer

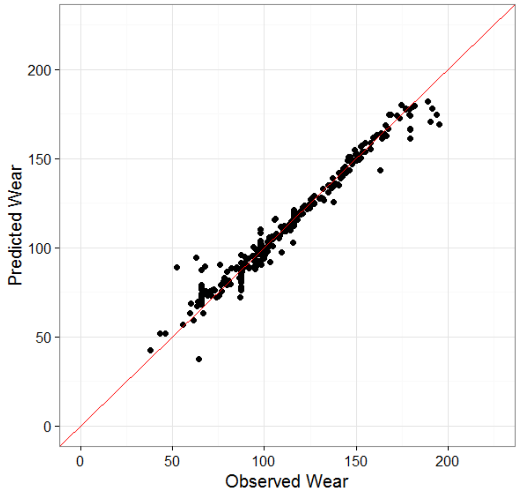

Figure 7 Comparison of observed and predicted tool wear using SVR

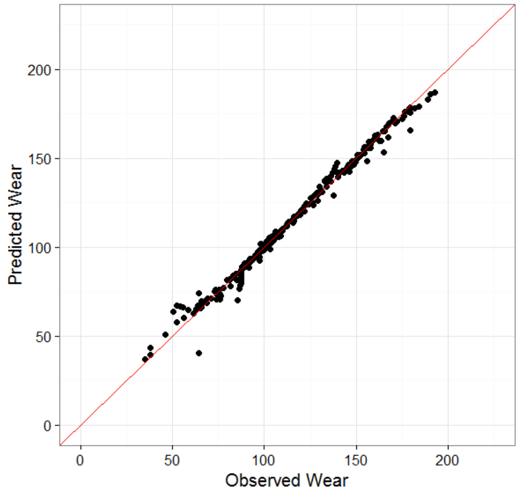

Figure 8 Comparison of observed and predicted tool wear using RFs

In addition, the performance of the three algorithms was evaluated using accuracy and training time. Accuracy is measured using the R^2 statistic, also referred to as the coefficient of determination, and mean squared error (MSE). In statistics, the coefficient of determination is defined as $R^2 = 1 - \frac{SSE}{SST}$ where SSE is the sum of the squares of residuals, SST is the total sum of squares. The coefficient of determination is a measure that indicates the percentage of the response variable variation that is explained by a regression model. A higher R-squared indicates that more variability is explained by the regression model. For example, an R^2 of 100% indicates that the regression model explains all the variability of the response data around its mean. In general, the higher the R-squared, the better the regression model fits the data. The MSE of an estimator measures the average of the squares of the errors. The MSE is defined as $MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$ where $\hat{y}_i$ is a predicted value, y_i is an observed value, and n is the sample size. The ANN, SVR, and RF algorithms use between 50% and 90% of the input data for model development (training) and uses the remainder for model validation (testing). Tables 4, 5, and 6 list the MSE, R-squared, and training time for the predictive models trained by the ANN, SRV, and RFs.

Table 4. Accuracy and Training Time for the FFBP ANN

	ANN (number of neurons = 16)		
Training size (%)	MSE	R^2	Training time (Second)
50	36.337	0.964	0.394
60	41.420	0.959	0.412
70	40.138	0.960	0.468
80	42.486	0.957	0.506
90	44.056	0.957	0.566

Table 5. Accuracy and Training Time for SVR

	SVR (Radial Basis Kernel)		
Training size (%)	MSE	R^2	Training time (Second)
50	54.993	0.946	0.060
60	49.868	0.952	0.073
70	41.072	0.959	0.088
80	31.958	0.969	0.107
90	23.997	0.975	0.126

Table 6. Accuracy and Training Time for RFs

	RFs (500 Trees)		
Training size (%)	MSE	R^2	Training time (Second)
50	14.170	0.986	1.079
60	11.053	0.989	1.386
70	10.156	0.990	1.700
80	8.633	0.991	2.003
90	7.674	0.992	2.325

RFs are the most accurate algorithm in comparison with ANNs and SVR based on MSE and R^2. However, the training time for RFs is longer than that of ANNs and SVR.

6. Conclusions and Future Work

In this paper, the prediction of tool wear in milling operations was conducted using three machine learning algorithms, including ANNs, SVR, and RFs. The performance of these algorithms was evaluated on the dataset collected from a milling experiment. The performance measures include mean squared error, r-squared, and training time. A set of statistical features was extracted from cutting forces, vibrations, and acoustic emissions. The experimental results have shown that while the training time on the particular dataset using RFs is longer than the FFBP ANNs with a single hidden layer and SVR, RFs generate more accurate predictions than the FFBP ANNs with a single hidden layer and SVR. The main contribution of this paper is twofold: (1) we demonstrated that the predictive model trained by RFs can predict tool wear in milling operations very accurately. To the best of our knowledge, the random forest algorithm is applied to predict tool wear in the manufacturing domain for the first time; (2) we compared the performance of RFs with that of ANNs and SVR, as well as observed that RFs outperform ANNs and SVR in this particular experiment.

In the future, a comparison of the performance of SVR and RFs with that of other types of ANNs, such as recurrent neural networks, will be conducted. In addition, our future work will focus on designing the parallel implementation of machine learning algorithms that can be applied to large-scale and real-time prognosis.

Acknowledgment

The research reported in this paper is partially supported by NSF under grant number IIP-1238335. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Swanson, L., 2001, "Linking Maintenance Strategies to Performance," International Journal of Production Economics, 70(3), pp. 237-244.

[2] Valdez-Flores, C., and Feldman, R. M., 1989, "A Survey of Preventive Maintenance Models for Stochastically Deteriorating Single-unit Systems," Naval Research Logistics (NRL), 36(4), pp. 419-446.

[3] Wu, D., Terpenny, J., Zhang, L., Gao, R., and Kurfess, T., 2016, "Fog-Enabled Architecture for Data-Driven Cyber-Manufacturing Systems," Proceedings of the ASME 2016 International Manufacturing Science and Engineering Conference, Blacksburg, Virginia, U.S.

[4] Lee, J., 1995, "Machine performance monitoring and proactive maintenance in computer-integrated manufacturing: review and perspective," International Journal of Computer Integrated Manufacturing, 8(5), pp. 370-380.

[5] Bevilacqua, M., and Braglia, M., 2000, "The Analytic Hierarchy Process Applied to Maintenance Strategy Selection," Reliab Eng Syst Safe, 70(1), pp. 71-83.

[6] Suh, J. H., Kumara, S. R., and Mysore, S. P., 1999, "Machinery fault diagnosis and prognosis: application of advanced signal processing techniques," CIRP Ann-Manuf Techn, 48(1), pp. 317-320.

[7] Hu, C., Youn, B. D., and Kim, T., 2012, "Semi-supervised learning with co-training for data-driven prognostics," Prognostics and Health Management (PHM), 2012 IEEE Conference on, IEEE, pp. 1-10.

[8] Schwabacher, M., 2005, "A Survey of Data-Driven Prognostics," Proceedings of the AIAA Infotech@ Aerospace Conference, Arlington, Virginia, pp. 1-5.

[9] Byrne, G., Dornfeld, D., Inasaki, I., Ketteler, G., König, W., and Teti, R., 1995, "Tool condition monitoring (TCM)—the status of research and industrial application," CIRP Ann-Manuf Techn, 44(2), pp. 541-567.

[10] Teti, R., Jemielniak, K., O'Donnell, G., and Dornfeld, D., 2010, "Advanced monitoring of machining operations," CIRP Ann-Manuf Techn, 59(2), pp. 717-739.

[11] Gao, R., Wang, L., Teti, R., Dornfeld, D., Kumara, S., Mori, M., and Helu, M., 2015, "Cloud-enabled prognosis for manufacturing," CIRP Ann-Manuf Techn, 64(2), pp. 749-772.

[12] Daigle, M. J., and Goebel, K., 2013, "Model-based Prognostics with Concurrent Damage Progression Processes," Systems, Man, and Cybernetics: Systems, IEEE Transactions on, 43(3), pp. 535-546.

[13] Si, X.-S., Wang, W., Hu, C.-H., Chen, M.-Y., and Zhou, D.-H., 2013, "A Wiener-process-based degradation model with a recursive filter algorithm for remaining useful life estimation," Mechanical Systems and Signal Processing, 35(1), pp. 219-237.

[14] Dong, M., and He, D., 2007, "Hidden semi-Markov model-based methodology for multi-sensor equipment health diagnosis and prognosis," European Journal of Operational Research, 178(3), pp. 858-878.

[15] Saha, B., Goebel, K., and Christophersen, J., 2009, "Comparison of prognostic algorithms for estimating remaining useful life of batteries," T I Meas Control.

[16] Orchard, M. E., and Vachtsevanos, G. J., 2009, "A particle-filtering approach for on-line fault diagnosis and failure prognosis," T I Meas Control.

[17] Wang, P., and Gao, R. X., 2015, "Adaptive resampling-based particle filtering for tool life prediction," Journal of Manufacturing Systems, 37, pp. 528-534.

[18] Sick, B., 2002, "On-line and indirect tool wear monitoring in turning with artificial neural networks: a review of more than a decade of research," Mechanical Systems and Signal Processing, 16(4), pp. 487-546.

[19] Kamarthi, S., Kumara, S., and Cohen, P., 2000, "Flank wear estimation in turning through wavelet representation of acoustic emission signals," Journal of Manufacturing Science and Engineering, 122(1), pp. 12-19.

[20] Liang, S., and Dornfeld, D., 1989, "Tool wear detection using time series analysis of acoustic emission," Journal of Engineering for Industry, 111(3), pp. 199-205.

[21] Huang, Y., and Liang, S. Y., 2004, "Modeling of CBN tool flank wear progression in finish hard turning," Journal of manufacturing science and engineering, 126(1), pp. 98-106.

[22] Taylor, F. W., 1907, On the art of cutting metals, American Society of Mechanical Engineers, New York.

[23] Schwabacher, M., and Goebel, K., 2007, "A survey of artificial intelligence for prognostics," Proc. AAAI fall symposium, pp. 107-114.

[24] Bukkapatnam, S. T., Lakhtakia, A., and Kumara, S. R., 1995, "Analysis of sensor signals shows turning on a lathe exhibits low-dimensional chaos," Phys Rev E, 52(3), p. 2375.

[25] Bukkapatnam, S. T., Kumara, S. R., and Lakhtakia, A., 2000, "Fractal estimation of flank wear in turning," Journal of Dynamic Systems, Measurement, and Control, 122(1), pp. 89-94.

[26] Bukkapatnam, S., Kumara, S., and Lakhtakia, A., 1999, "Analysis of acoustic emission signals in machining," Journal of Manufacturing Science and Engineering, 121(4), pp. 568-576.

[27] Özel, T., and Karpat, Y., 2005, "Predictive modeling of surface roughness and tool wear in hard turning using regression and neural networks," International Journal of Machine Tools and Manufacture, 45(4), pp. 467-479.

[28] Palanisamy, P., Rajendran, I., and Shanmugasundaram, S., 2008, "Prediction of tool wear using regression and ANN models in end-milling operation," The International Journal of Advanced Manufacturing Technology, 37(1-2), pp. 29-41.

[29] Sanjay, C., Neema, M., and Chin, C., 2005, "Modeling of tool wear in drilling by statistical analysis and artificial neural network," J Mater Process Tech, 170(3), pp. 494-500.

[30] Chungchoo, C., and Saini, D., 2002, "On-line tool wear estimation in CNC turning operations using fuzzy neural network model," International Journal of Machine Tools and Manufacture, 42(1), pp. 29-40.

[31] Chen, J. C., and Chen, J. C., 2005, "An artificial-neural-networks-based in-process tool wear prediction system in milling operations," The International Journal of Advanced Manufacturing Technology, 25(5-6), pp. 427-434.

[32] Cho, S., Asfour, S., Onar, A., and Kaundinya, N., 2005, "Tool breakage detection using support vector machine learning in a milling process," International Journal of Machine Tools and Manufacture, 45(3), pp. 241-249.

[33] Benkedjouh, T., Medjaher, K., Zerhouni, N., and Rechak, S., 2015, "Health assessment and life prediction of cutting tools based on support vector regression," Journal of Intelligent Manufacturing, 26(2), pp. 213-223.

[34] Jiaa, C. L., and Dornfeld, D. A., 1998, "A self-organizing approach to the prediction and detection of tool wear," ISA transactions, 37(4), pp. 239-255.

[35] Elangovan, M., Devasenapati, S. B., Sakthivel, N., and Ramachandran, K., 2011, "Evaluation of expert system for condition monitoring of a single point cutting tool using principle component analysis and decision tree algorithm," Expert Systems with Applications, 38(4), pp. 4450-4459.

[36] McCulloch, W. S., and Pitts, W., 1943, "A logical calculus of the ideas immanent in nervous activity," The bulletin of mathematical biophysics, 5(4), pp. 115-133.

[37] Cortes, C., and Vapnik, V., 1995, "Support-vector networks," Machine learning, 20(3), pp. 273-297.

[38] Smola, A., and Vapnik, V., 1997, "Support vector regression machines," Advances in neural information processing systems, 9, pp. 155-161.

[39] Smola, A. J., and Schölkopf, B., 2004, "A tutorial on support vector regression," Statistics and computing, 14(3), pp. 199-222.

[40] Liaw, A., and Wiener, M., 2002, "Classification and regression by randomForest," R news, 2(3), pp. 18-22.

[41] Breiman, L., 2001, "Random forests," Machine learning, 45(1), pp. 5-32.

[42] Friedman, J., Hastie, T., and Tibshirani, R., 2001, The elements of statistical learning, Springer series in statistics Springer, Berlin.

[43] Heaton, J., 2008, Introduction to neural networks with Java, Heaton Research, Inc.

[44] Blum, A., 1992, Neural networks in C++: an object-oriented framework for building connectionist systems, John Wiley & Sons, Inc.

[45] Li, X., Lim, B., Zhou, J., Huang, S., Phua, S., Shaw, K., and Er, M., 2009, "Fuzzy neural network modelling for tool wear estimation in dry milling operation," Annual Conference of the Prognostics and Health Management Society, pp. 1-11.