Indirect Tool Condition Monitoring Using Ensemble Machine Learning Techniques

Alexandra Schueller[1]
George W. Woodruff
School of Mechanical Engineering,
Georgia Institute of Technology,
801 Ferst Drive,
Atlanta, GA 30332
e-mail: aschueller3@gatech.edu

Christopher Saldaña
George W. Woodruff
School of Mechanical Engineering,
Georgia Institute of Technology,
801 Ferst Drive,
Atlanta, GA 30332
e-mail: christopher.saldana@me.gatech.edu

Tool condition monitoring (TCM) has become a research area of interest due to its potential to significantly reduce manufacturing costs while increasing process visibility and efficiency. Machine learning (ML) is one analysis technique which has demonstrated advantages for TCM applications. However, the commonly studied individual ML models lack generalizability to new machining and environmental conditions, as well as robustness to the unbalanced datasets which are common in TCM. Ensemble ML models have demonstrated superior performance in other fields, but have only begun to be evaluated for TCM. As a result, it is not well understood how their TCM performance compares to that of individual models, or how homogeneous and heterogeneous ensemble models' performances compare to one another. To fill in these research gaps, milling experiments were conducted using various cutting conditions, and the model groups were compared across several performance metrics. Statistical t-tests were also used to evaluate the significance of model performance differences. Through the analysis of four individual ML models and five ensemble models, all based on the processes' sound, spindle power, and axial load signals, it was found that on average, the ensemble models performed better than the individual models, and that the homogeneous ensembles outperformed the heterogeneous ensembles. [DOI: 10.1115/1.4055822]

Keywords: machine learning, machining, process monitoring, tool condition monitoring, tool wear, machining processes, production systems optimization, sensing, monitoring, and diagnostics

1 Introduction

To meet industry goals of increasing manufacturing efficiency while reducing costs and cycle times, businesses have been turning to automation and techniques for increasing process visibility to optimize operations. One method of achieving this during metal machining processes is through tool condition monitoring (TCM). TCM systems, in which the user is provided with information regarding a tool's wear level during a machining process, have the potential to significantly improve these manufacturing metrics. For example, a TCM system's prevention of the continued use of very worn tools would provide final products with better surface finish, dimensional accuracy, fatigue strength, and other material properties [1–7]. This would also allow processes to be completed with decreased power consumption, process temperatures, and machine vibration [6]. An effective TCM system would also allow many companies to turn from their conservatively estimated tool change times to times chosen based on real-time condition data for the individual tool in use, thereby decreasing, on average, the amount of tooling and tool changing time necessary to complete a product [3,8–10].

While several research groups have studied this issue and employed various data collection and analysis methods to address turning, milling, drilling, and grinding processes, a practical and reliable TCM system for industry machining has yet to be offered [9,10,12–14]. Sensor fusion, in which data from multiple sensor types are used by an algorithm, is one advancement which has improved TCM systems by reducing uncertainty, minimizing sensitivity to sensor noise, and improving overall algorithm performance [2,7,10,11,14–18].

Similarly, machine learning (ML) is an analysis technique which, in recent decades, has proven to be a valuable tool for TCM. Its advantages in large-scale data processing, autonomous nonlinear pattern recognition, and multivariate analysis have enabled the significant improvement of TCM systems' performance, automatization, and adaptability to new conditions [6,17–20]. While many individual ML models have been applied and compared for TCM, it remains difficult for any single learner to achieve high classification accuracies while also avoiding overfitting and keeping generalization ability high [14,21]. Ensemble ML, a technique that utilizes the collective knowledge and insights generated by multiple base learners to create an improved final model, has recently been shown in limited cases to further improve TCM system performance [15,21–25].

Ensemble ML has been applied to TCM problems using surface texture images [23], spindle motor current and power [22], vibration [24], acoustic emission [15], and cutting force signals [21,25]. However, it is not well understood how various types of TCM ensemble ML models' performance compares to that of standalone ML models in terms of accuracy, generalizability, and robustness to the unbalanced class sizes which are common in TCM experiments when data from several individual tools and machining conditions are used, and sensor fusion of practical sensors is focused on. Many studies are limited by their use of data from the same experiment in both their training and testing sets, which significantly limits their ability to study model generalizability, while others use intrusive sensors like dynamometers and acoustic emission sensors which are less practical for use in industrial environments. Further, research gaps exist surrounding the performance differences of heterogeneous and homogeneous ensemble models for TCM, such as how certain ensemble models such as extra-trees and soft voting perform for signal-based wear monitoring, how they compare across various machining condition changes, and how robust they are to the class imbalance in TCM.

[1]Corresponding author.

Manuscript received June 30, 2022; final manuscript received September 22, 2022; published online October 13, 2022. Assoc. Editor: Andy Henderson.

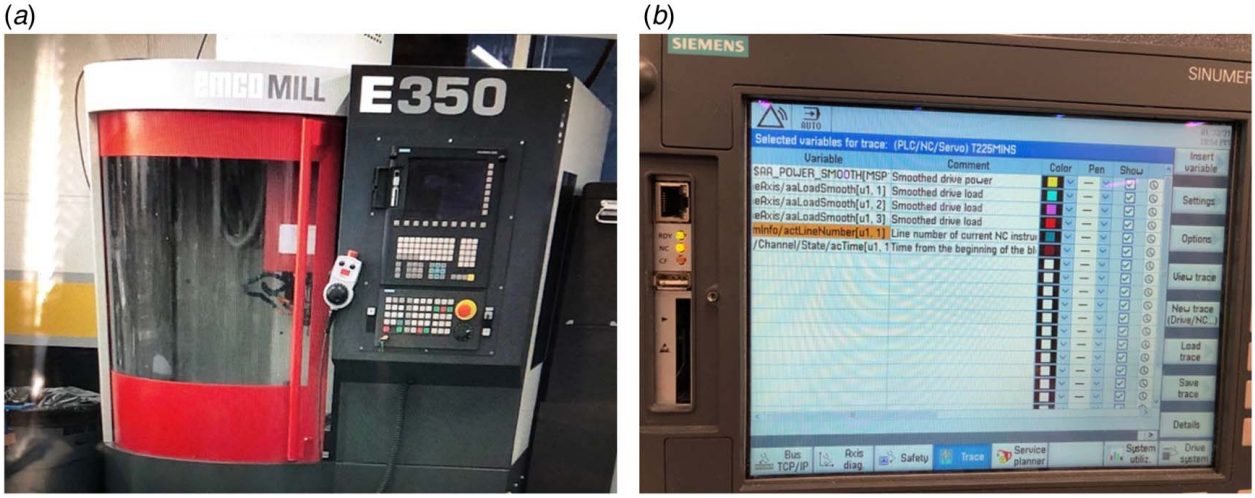

Fig. 1 (*a*) EMCOMILL E350 three-axis vertical CNC mill, equipped with (*b*) Siemens 828D controller

Heterogeneous ensembles are made up of base learners of completely different configurations, while homogenous ensembles are made up of learners of the same configuration, but which may use different initiation parameters, weight values, or training data subsets. These two types of ensemble ML models offer unique advantages. Heterogeneous ensembles draw upon the insights of diverse base learners while attempting to outweigh each learner's drawbacks. Homogeneous ensembles reduce overfitting and reliance on dataset selection through the employment of larger numbers of randomized base learners. It is important to understand how the groups' performance compares for this application.

In order to fill the research gaps surrounding the understanding of how an ML model's configuration type affects its performance for TCM across metrics such as classification accuracy, accuracy standard deviation (SD), robustness to unbalanced datasets, and computation time, an experimental study was designed. Sound, spindle power, and axial load data were collected from tool life experiments, and ML models were trained using features from across the different process signals. Model performances were evaluated using different cross validation (CV) approaches, confusion matrices, and tests using various splits of the experiments' data. The results obtained were used to identify the advantages and disadvantages of individual, ensemble, heterogeneous ensemble, and homogeneous ensemble ML models for TCM.

2 Methods

2.1 Tool Wear Experiments and Data Collection. Tool life experiments used to gather sufficient data for this study, were run on an EMCOMILL E350 three-axis vertical CNC mill equipped with a Siemens 828D controller, as shown in Fig. 1. Spindle power and the three axial load signals were sampled internally at 166 Hz through the controller, and the sound signal was sampled at 44.1 kHz using an external microphone installed at the top of the machine. A Model 485B39 Digital ICP signal conditioner and the MATLAB audio labeler application were also used to process and collect the sound data from the microphone, and an analog sound calibrator was used to check the sound signals' accuracy. Finally, a Dino-Lite AF3113T microscope was also installed inside the machine to capture intermittent images of the tool's condition.

Table 1 Experimental overview

Experiment #	Parameter set	Spindle speed	Feed rate
1	A	HI	HI
2	B	HI	LO
3	C	LO	HI
4	D	LO	LO
5	D	LO	LO
6	C	LO	HI
7	B	HI	LO
8	A	HI	HI

Each experiment consisted of an initially new end mill insert being used to cut straight passes across a $6 \times 1 \times 2$ in. ($15.24 \times 2.54 \times 5.08$ cm) D2 steel workpiece, until it reached a maximum measured flank wear (VB_{max}) of 0.3 mm (0.012 in.). This wear threshold is in accordance with ISO8688-2 [26], as well as many earlier studies, and allows the ML models to focus on a wear range which would be seen often in industrial manufacturing. Images of the tool were taken after each set of three passes, and a calibration scale was used to confirm image dimensions during the manual measurement of its maximum wear along the flank surface. The tooling used for the experiments were 0.5 in. (1.27 cm) diameter one-flute Kennametal KICR end mill bodies, equipped with square-shaped TiAlN-PVD-coated carbide Kennametal KIPR inserts designed for milling steel.

The eight experiments were run using four distinct machining parameter sets, as given in Table 1. The experimentation conditions consisted of two spindle speeds, 4400 rpm (high setting; HI) and 3700 rpm (low setting; LO); two feed rates, 34 ipm (HI) and 24 ipm (LO); and two identical repetitions of each parameter set. The other machining parameters, including an axial depth of cut of 3.81 mm (0.15 in.) and a radial depth of cut of 0.635 mm (0.025 in.), were kept constant. Water-based coolant was used for all experiments.

2.2 Data Pre-Processing. The data collected from the experiments were processed using the PYTHON IDE PYCHARM, to become usable as training and testing data for the ML algorithms. The portions of each machining pass in which the tool's center was more than one tool radius from either end of the material were identified, and that data were split into 1 s segments. Each of these segments, which included a 1 s sample from each of the five process signals, served as an independent sample for the ML analysis, and there was no overlap between samples. A 1 s time frame for each ML data point was selected due to its use in prior research, its ability to provide a satisfactory number of data points for the study, and the relatively slow progression of tool wear which makes a higher prediction frequency unnecessary.

An individual flank wear measurement was calculated for each 1 s sample using linear interpolation between each pair of wear images, and the segment's discrete wear class was labeled using

Table 2 Tool flank wear (VB) measurements for the three wear levels

Wear level	VB range (mm)	VB range (in.)
1	0.0–0.1	0.000–0.004
2	0.1–0.2	0.004–0.008
3	0.2–0.3	0.008–0.012

Table 3 Selected features using RFECV

Signal type	Selected features
Sound	TD mean
	TD standard deviation
	TD skewness
	TD median
	FD amplitude sum of 0–9 kHz
Spindle power	TD mean
	TD RMS
X-axis load	TD standard deviation
	TD skewness
Y-axis load	TD mean
	TD maximum
	TD clearance factor
Z-axis load	TD mean
	TD median

the wear ranges shown in Table 2. A prediction resolution of three wear levels was chosen in order to give a higher resolution than with only two classes, while allowing the results of this study to be comparable to the prior works which have used three wear levels. These were the classes which the ML models aimed to label correctly for each sample of data.

As images of the tools' wear were only taken after each set of three machining passes, it was assumed that within each of these machining sets, the maximum flank wear measurement progressed linearly. This is a sufficiently valid assumption for this application due to the consistent cutting conditions being used and flank wear's generally progressive nature within short time frames, especially during the initial wear and normal wear stages observed most in this study. The lack of any chipping being observed during the experiments also supports this applicability. In addition, the study does not rely on every sample's wear measurement being calculated exactly—it only relies on the samples being correctly classified into the three wear levels. Only the samples from the three machining passes over which each tool transitioned from wear level 1 to level 2, and from level 2 to level 3 (10.6% of the data) were at any risk of being misclassified due to this assumption.

A total of 79 features were extracted from the time and frequency domains of the samples' signals, based on features found to be highly correlated to tool wear or related conditions by earlier works [2,7,14–16,18,22,27–35]. More detail on the full list of features studied and the equations used for this are described in Ref. [36]. The distributions of these features were standardized using a mean of zero and a standard deviation of one to avoid prediction bias due to varying feature scales. The standardization method was used for this over other approaches due to its low level of vulnerability to outliers in the data. Then, a feature selection method called recursive feature elimination using cross validation (RFECV) and a support vector machine (SVM) ML model were applied to reduce the feature set down to the 14 features which provide the largest contribution to TCM model performance. The RFECV method showed that a feature subset of the 14 features listed in Table 3 provided the strongest SVM model cross validation performance out of the many evaluated feature combinations and subset sizes. This selected feature subset included features from all five process signals, showing that all five signals provided

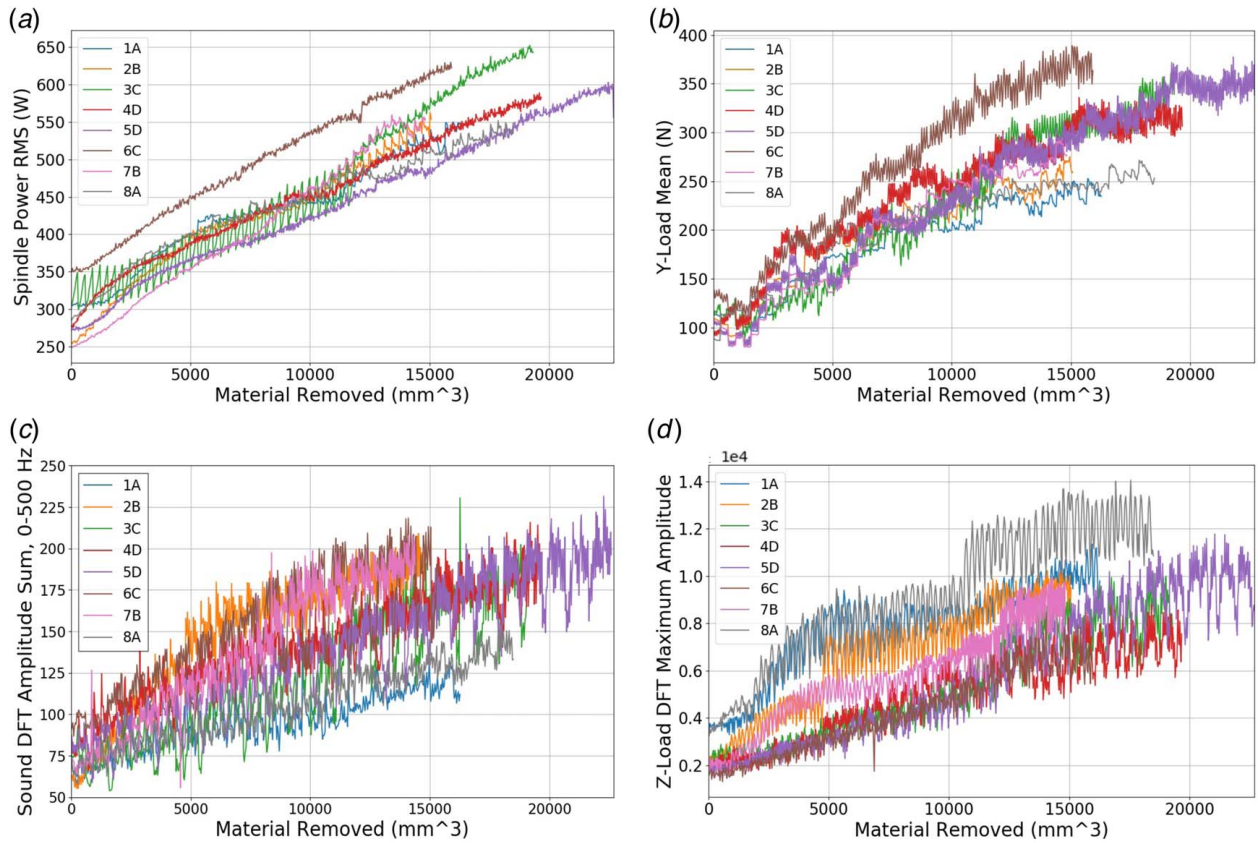

Fig. 2 Change in (*a*) spindle power RMS, (*b*) *y*-axis mean axial load, (*c*) sound DFT amplitude sum across frequency range 0–500 Hz, and (*d*) *z*-axis load DFT maximum amplitude, with increasing material removed

Table 4 Summary of collected experimental data

Experiment label	# Passes to flank wear cut off	Total material removed (mm^3)	Total material removed ($in.^3$)	Total machining time (min)	Total # ML samples	Samples in wear level 1	Samples in wear level 2	Samples in wear level 3
1A	54	19,910	1.215	9.5	468	36	126	306
2B	49	18,070	1.102	12.3	614	64	269	281
3C	59	21,750	1.327	10.4	526	54	304	168
4D	63	23,230	1.417	15.8	801	78	406	317
5D	73	26,920	1.642	18.3	924	76	439	409
6C	47	17,330	1.057	8.3	422	34	229	159
7B	48	17,700	1.080	12.0	604	99	256	249
8A	59	21,750	1.327	10.4	527	40	250	237
Total	452	166,660	10.167	96.9	4,886	481	2,279	2126

useful information about the tools' wear. Performing this feature reduction allowed the system's computation time to be reduced, as well as prevented unrelated features from distracting the algorithm from features that were more highly correlated to tool wear.

The collected data for four features that were studied are shown in Fig. 2. The root mean square (RMS) of the spindle power, shown in Fig. 2(*a*), and the mean *y*-axis load, shown in Fig. 2(*b*) were selected for use by the RFECV algorithm, while the sound discrete Fourier transform (DFT) amplitude sum over the range of 0–500 Hz, shown in Fig. 2(*c*), and the *z*-axis load DFT maximum amplitude, shown in Fig. 2(*d*) were not selected. It can be observed that the chosen features generally showed less fluctuation with increasing material removed (MR), as well as a more consistent upward trajectory between the different experiments.

For each experiment, half of the samples were set aside for model validation, while the other half was used for model evaluation. The data were split using stratified random sampling to ensure that the data from each experiment and wear level were split evenly between the two groups. The ML model parameter tuning and feature selection were done using 20x-repeated 10-fold cross validation on the validation dataset. Then, the evaluation dataset was split into training and testing groups based on the relevant machining conditions for each analysis, as shown in Table 8. It was beneficial to split the data into the validation and evaluation groups first in order to ensure that no overlap occurred between them, while allowing several different training/testing splits of the evaluation dataset to be applied for the various analyses covered in the study. The data to be used for the various analyses' training and testing sets were standardized before each ML run.

2.3 Machine Learning Models. To compare the overall performance of individual, ensemble, heterogeneous ensemble, and homogeneous ensemble ML models for TCM, nine different ML algorithms were analyzed using version 0.22.2. post1 of the scikit-learn PYTHON library. Four individual models were applied: decision tree (DT), SVM, k nearest neighbors (kNN), and artificial neural network (ANN). The SVM model, more specifically, was a multiclass SVM based on a one-versus-one scheme, and the ANN model was a multilayer perceptron. These algorithms were chosen based on their history of performing better than other individual models for TCM [14,20,21,28,29,37–41], and for their diversity of decision strategies which could provide a range of advantages for ensemble models to benefit from. Five ensemble ML models were studied: three heterogeneous ensembles and two homogeneous ensembles. The three heterogeneous ensemble models included ensemble hard voting (EHV), ensemble soft voting (ESV), and a stacked generalization model using an SVM meta-learner (stacked SVM). These were all constructed using the four individual models listed earlier. The two homogeneous ensembles included random forest (RF) and extremely randomized trees ("extra-trees" (ET)). These were both constructed using a "forest" of decision trees. The ensemble models were selected based on the EHV, stacked SVM, and RF models' past successes in the limited number of studies completed in this area [15,21–25], as well as on the ESV and ET models' strong performance in related areas but lack of prior application to signal-based TCM [23,42–46].

Extensive grid searches were performed using the validation dataset and 10-fold cross validation, in order to tune each model's hyperparameters and optimize their TCM performance. To perfect the decision tree, the maximum number of features considered at a time was set to "None", the minimum number of samples needed for an internal node to split was set to two, no maximum tree depth was set, and a Gini impurity criterion was selected for measuring the node split quality. Through tuning the SVM, a radial basis function kernel was selected and the regularization parameter is set at $C = 5$. The number of neighbors considered by the kNN model was set at $k = 3$, and Euclidean distance measures and uniform sample weights were used. To optimize the ANN model, a rectified linear unit function was chosen for the activation function, a stochastic gradient-based optimizer was used for the weight optimization parameter, a value of 0.0001 was chosen for the L2 regularization parameter, and the hidden layer sizes were set to 6, 30, and 37. The classifier weights for the EHV and ESV models were set according to their base models' 10-fold CV scores using the training data, which allowed ties between the four base models' votes to be avoided and gave slightly more weight to better-performing base models. The SVM meta-learner for the stacked SVM model used no classifier weights and a regularization parameter of $C = 1$. Finally, a forest of 100 trees was selected for the RF and ET homogeneous ensembles.

3 Results and Discussion

3.1 Collected Data. An overview of the data collected during the tool wear experiments is shown in Table 4. A total of 4886 original milling data samples, each including 1 s segments from all five process signals, were collected and split according to the procedure as described in Sec. 2.2. Roughly 10% of these samples were collected while the tool was in tool wear level 1, 47% are from wear level 2, and 43% are from wear level 3. This resulted in an unbalanced dataset, in which the amount of data available was not evenly distributed between the classes used. For the TCM application, this is common due to the changing wear-rates present during different stages of a tool's life, and in particular, the very high wear-rate during a tool's initial wear stage.

In Fig. 3, sample microscope images from tool wear levels (*a*) 1, (*b*) 2, and (*c*) 3 are shown. The tool insert's condition deterioration and flank wear progression can be observed on the left side of the insert as the amount of MR increased.

3.2 Evaluation of Machine Learning Model Configurations. ML model performance was evaluated in several ways using the evaluation dataset described in Sec. 2.2. Figure 4

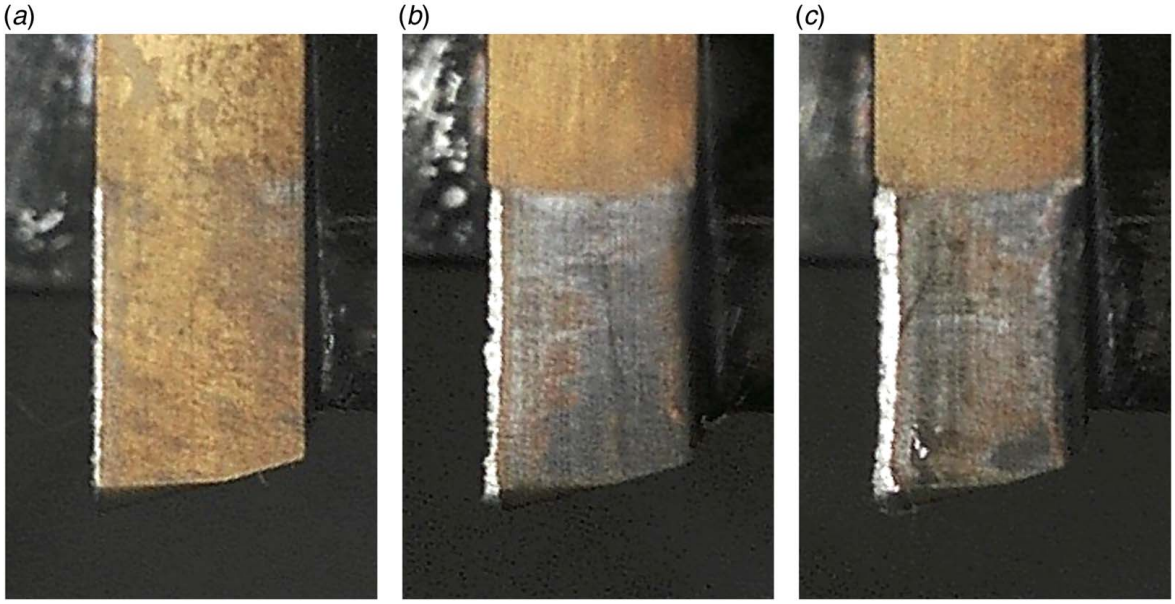

Fig. 3 Tool condition deterioration during experiment 4D, shown at (*a*) MR = 940 mm^3 (0.057 in.^{3}) and VB_{max} = 0.074 mm (0.003 in.), (*b*) MR = 8460 mm^3 (0.516 in.^{3}) and VB_{max} = 0.168 mm (0.007 in.), and (*c*) MR = 19,100 mm^3 (1.166 in.^{3}) and VB_{max} = 0.287 mm (0.011 in.)

shows the individual, ensemble, heterogeneous ensemble, and homogeneous ensemble model groups' averaged accuracies using 20 times repeated (*a*) 10-fold CV, and (*b*) leave-one-group-out cross validation (LOGO-CV). While 10-fold CV is commonly used in TCM system research and many other applications, LOGO-CV has an advantage in that it allows the user to manually select the data which are used for each CV group. For TCM, this can ensure that during the cross validation process, different data samples from the same experiment would never be used in both the training and testing ML sets. This gives a more accurate measure of how a real-world TCM classification model would perform.

Based on Fig. 4, it can be observed that when the accuracies for all of the models in each group were averaged, the ensemble model group performed better than the individual model group. In addition, the homogeneous ensemble group performed better than the heterogeneous ensemble group. These patterns were consistent across both the 10-fold CV and LOGO-CV metrics. Using 10-fold CV, the ensemble model group gave an average classification error which was 28.7% lower than that of the individual model group, and using LOGO-CV, the ensemble group's error was 8.5% lower than that of the individual model group. When the homogeneous ensemble models were compared with the heterogeneous ensemble models, the homogeneous ensemble group achieved an average classification error which was 44.5% lower than that of the heterogeneous ensemble group using 10-fold CV, and 8.3% lower using LOGO-CV. Tables 5 and 6 summarize the statistical analysis of model groups' accuracy differences using *t*-tests. Using a cutoff *p*-value of 0.05 and either 10-fold CV scores for Table 5, or LOGO-CV scores for Table 6, the differences between the homogeneous ensemble group's accuracies and those of the other models were all found to be statistically significant. In addition, the 10-fold CV and LOGO-CV accuracies for each of the nine models studied, shown in Table 7, confirm the results in Fig. 4 by showing that the extra-trees and random forest models scored the highest in 10-fold CV and LOGO-CV, respectively.

A set of 15 generalizability tests, as shown in Table 8, were also run in order to evaluate the different model types' performance across various machining condition changes. Performance metrics are reported for these tests in two groups: one is based on analyses 2–6, which evaluate the models' generalizability to condition changes, and one is based on analysis 1 which shows the models' performance on testing data from the same experiment. These

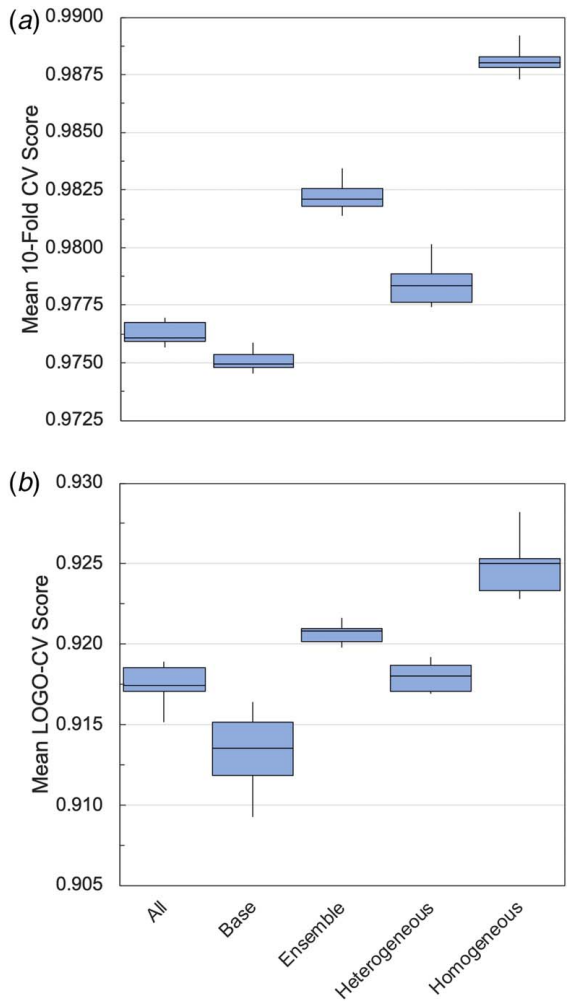

Fig. 4 Average accuracies of base, ensemble, heterogeneous ensemble, and homogeneous ensemble ML model groups, using (*a*) 10-fold cross validation and (*b*) leave-one-group-out cross validation

Table 5 Statistical analysis of model group performance comparisons using 10-fold CV scores

Model 1: Best-performing group using 20x-repeated 10-fold CV	Homogeneous ensemble models		
Model 2: Comparison group	Base models	Ensemble models	Heterogeneous models
t-value	68.89	25.04	32.79
p-value	3.76×10^{-28}	4.01×10^{-17}	1.60×10^{-19}
p-value < 0.05?	Yes	Yes	Yes

Table 6 Statistical analysis of model group performance comparisons using LOGO-CV scores

Model 1: Best-performing group using 20x-repeated LOGO-CV	Homogeneous ensemble models		
Model 2: Comparison group	Base models	Ensemble models	Heterogeneous models
t-value:	15.10	7.33	11.68
p-value:	1.98×10^{-13}	3.22×10^{-7}	1.19×10^{-10}
p-value < 0.05?	Yes	Yes	Yes

Table 7 10-fold and leave-one-group-out cross validation accuracies for all models

	ML models								
Performance metric	DT	SVM	kNN	ANN	EHV	ESV	Stacked SVM	RF	ET
10-fold CV mean accuracy	0.975	0.966	0.987	0.972	0.985	0.985	0.965	0.987	0.989
10-fold CV accuracy SD	0.002	0.001	0.001	0.001	0.001	0.001	0.002	0.001	0.001
LOGO-CV mean accuracy	0.901	0.917	0.910	0.922	0.919	0.919	0.916	0.926	0.924
LOGO-CV accuracy SD	0.007	0.000	0.000	0.007	0.002	0.002	0.002	0.002	0.002

metrics were averaged for each model group and are listed in Table 9. The accuracy, recall, precision, and F1 score are reported using macro-averaging and weighted averaging, two methods of generating one representative score for classification problems of more than two classes. For TCM, some of the most important metrics within those reported are the weighted and macro-averaged F1 score for analyses 2–6 of the generalizability study, and the leave-one-group-out CV accuracy. This is due to these scores' focus on evaluating the models on new machining conditions, similar to how they would be used in a real-world environment, as well as the F1 scores' balance of recall and precision.

Table 8 ML condition variability analyses

Analysis	Training data	Testing data
1 Data taken from same experiment	1A set 1 2B set 1 3C set 1 4D set 1	1A set 2 2B set 2 3C set 2 4D set 2
2 New tool, but same machining parameters used	1A 2B 3C 4D	8A 7B 6C 5D
3 Feed rate changed	2B, 7B 4D, 5D	1A, 8A 3C, 6C
4 Spindle speed changed	3C, 6C 4D, 5D	1A, 8A 2B, 7B
5 Feed rate and spindle speed changed	1A, 8A 2B, 7B	4D, 5D 3C, 6C
6 Mix of all parameter sets	1A, 2B, 3C, 4D	5D, 6C, 7B, 8A

Additionally, any differences between the weighted and macro-averaged scores can provide information about the models' robustness to unbalanced datasets.

From these results, with the highest scores highlighted in dark shade in the table, it is clear that across the majority of the metrics presented, the homogeneous ensemble group performed the best. However, the two metrics in which the heterogeneous ensemble group scored highest, including the generalizability runs' average accuracy and weighted F1 score, also give insight into the model groups' differences. As the heterogeneous ensemble group achieved the highest accuracy (also known as weighted recall), but a significantly lower macro-averaged recall, this suggests that it usually performed sub-optimally on the data from tool wear level 1, the level containing the smallest number of true samples. As macro-averaging equalizes the weights of the wear classes, this would increase the importance of the first wear level's data in the recall metric and have the observed effect. Then, the models' weighted F1 scores are dependent on these weighted recall scores. These recall score differences, especially compared to the less distinct differences for the homogeneous ensemble group, suggest that the heterogeneous ensemble models showed lower effectiveness for unbalanced datasets than the homogeneous ensemble models. The differences in heterogeneous and homogeneous ensemble model performance could be due to the variation in base predictions and insights gained through the homogeneous models' process of training the base models on different randomized subsets of the available data, or the homogeneous models' randomization of some base model parameters, both of which can reduce model overfitting. The homogeneous models were also able to use a higher number of base models due to decision trees' low computation complexity, which may allow them to benefit from a higher number of learned patterns in the data.

Table 9 Model group comparison across various performance metrics

Analysis category	Performance metric	Individual ML models (DT, SVM, kNN, and ANN)	All ensembles (EHV, ESV, Stacked SVM, RF, and ET)	Heterogeneous ensembles (EHV, ESV, and Stacked SVM)	Homogeneous ensembles (RF and ET)
Generalizability study (analyses 2–6)	Mean accuracy (weighted recall)	0.857	0.876	0.881	0.868
	Accuracy SD	0.021	0.013	0.018	0.006
	Mean accuracy 95% confidence interval	0.013	0.008	0.011	0.004
	Macro-averaged recall	0.775	0.783	0.772	0.800
	Weighted precision	0.884	0.899	0.895	0.903
	Macro-averaged precision	0.820	0.854	0.840	0.874
	Weighted F1 score	0.841	0.863	0.867	0.857
	Macro-averaged F1 score	0.820	0.860	0.850	0.874
Same experiment (analysis 1)	Mean accuracy	0.975	0.981	0.978	0.986
	Accuracy SD	0.003	0.003	0.004	0.002
Cross validation	10-fold CV mean accuracy	0.975	0.982	0.978	0.988
	10-fold CV accuracy SD	0.001	0.001	0.001	0.001
	LOGO-CV mean accuracy	0.912	0.921	0.918	0.925
	LOGO-CV accuracy SD	0.003	0.002	0.002	0.002

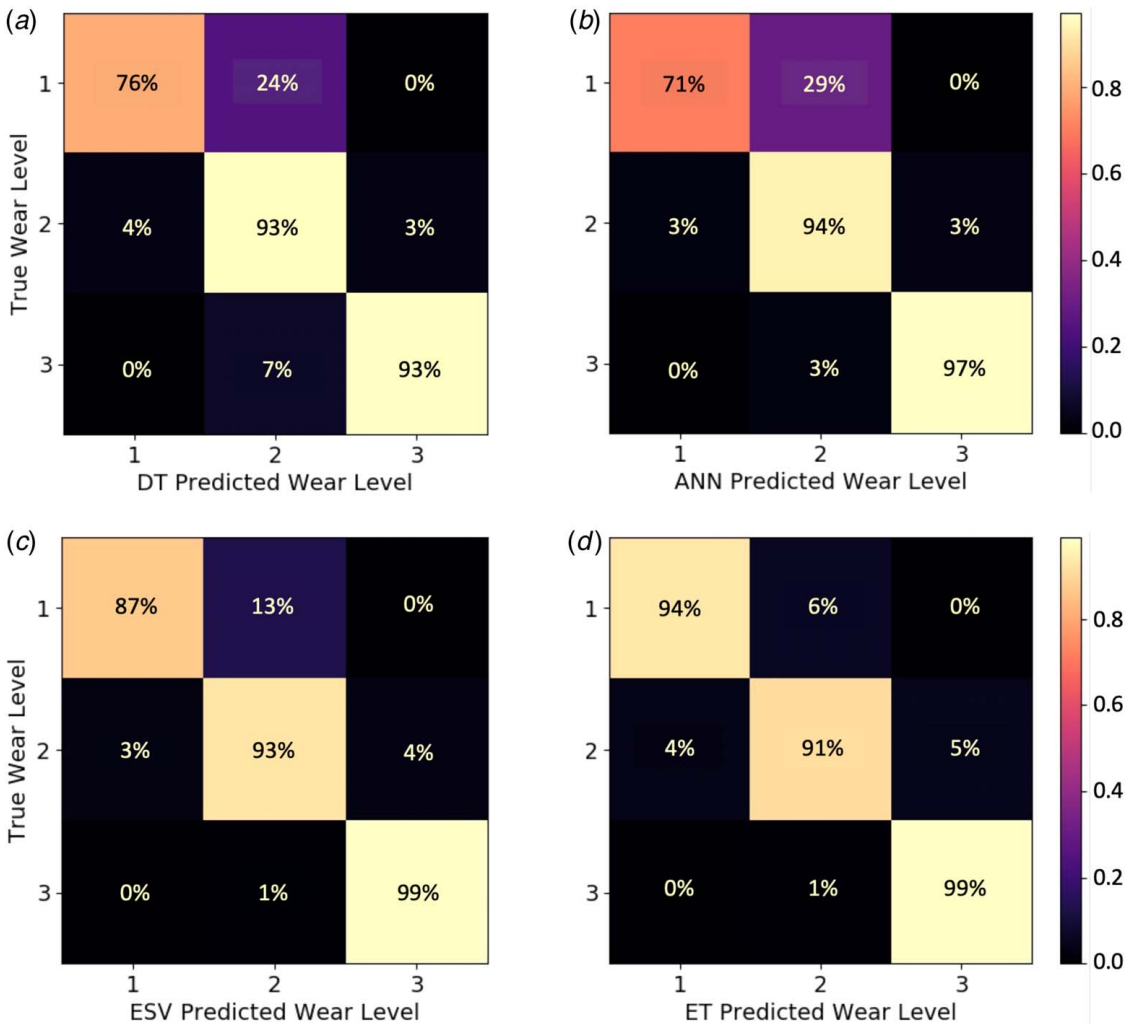

Fig. 5 Confusion matrices for (*a*) the decision tree, (*b*) the artificial neural network, (c) the soft voting ensemble, and (*d*) the extra-trees ensemble models, using the evaluation data from experiments 1–4 as the training set and the evaluation data from experiments 5–8 as the testing set

In addition, across all of the metrics, the ensemble model group outperformed the individual model group. Based on the average generalizability study accuracy scores, the ensemble models reduced the TCM classification error by 13.2% compared to the individual models. Based on analysis 1, the ML runs in which data from the same experiment were used in both the training and testing sets, this reduction is by 25.0%. The improved performance of the ensemble models for TCM compared to the individual models is expected due to their improved model accuracy and generalizability in other research areas, as well as their ability to make more-informed predictions based on the patterns detected by all of their base models instead of only one, but the levels and details of the improvement measured in this study are valuable at a time when limited research has been conducted into ensemble ML for TCM.

Finally, the standard deviations of the four accuracy scores reported were generally reduced when the ML model type was changed from individual to the ensemble, or heterogeneous ensemble to homogeneous ensemble. This reduction in accuracy standard deviation between the individual and ensemble models is expected due to the individual models' singular sets of insights and limitations, while the ensemble models can balance some base models' knowledge gaps with the insights from other base models. In addition, the reduction in accuracy standard deviation between the heterogeneous and homogeneous ensemble models is also expected due to the homogeneous models' more similar base models and therefore base model predictions, as well as the larger number of base models used by the homogeneous models in this study.

Figure 5 shows confusion matrices for the decision tree, neural network, soft voting, and extra-trees models. By using data from experiments 1–4 as the training set and data from experiments 5–8 as the testing set, they aim to provide a good average assessment of how models perform when they are trained on several different machining conditions and tested on data from a new individual tool.

As shown in Fig. 5, the DT, ANN, and ESV models all showed lower accuracies for wear level 1 than they did for the higher wear levels. In addition, between levels 1 and 2, they all showed bias toward level 2, the larger class, as shown by the significant error found in the top-middle boxes of Fig. 5. Meanwhile, the extra-trees model's accuracies were relatively constant across all three wear levels, and no significant bias was identified in its classifications between levels 1 and 2. These results, along with the similar confusion matrices of the other models, suggest a higher level of robustness to unbalanced class sizes for the ensemble models compared to the individual models, as well as for the homogeneous ensembles compared to the heterogeneous ensembles. These results confirm those described in Table 9.

Finally, Tables 10 and 11 show the computation time results for each model, as well as the mean results for each model group. These values were calculated by averaging the computation times from five repetitions, all using a computer with a 1.6 GHz processor. Since the most relevant time constraint for TCM is that the system must be able to take in signal data, calculate features, and make a classification decision within the same time duration of each data sample in order for real-time monitoring to be effective, these computation time components are shown.

Table 10 Computation times for one sample, using each model

ML model	Time to calculate sound features (ms)	Time to calculate controller features (ms)	Average ML classification time (ms)	Total in-situ computation time per sample (ms)
DT	195.45	0.96	0.00	196.41
SVM	195.45	0.96	0.02	196.43
kNN	195.45	0.96	0.07	196.48
ANN	195.45	0.96	0.00	196.41
EHV	195.45	0.96	0.12	196.53
ESV	195.45	0.96	0.05	196.46
Stacked SVM	195.45	0.96	0.07	196.48
RF	195.45	0.96	0.03	196.44
ET	195.45	0.96	0.03	196.44

Table 11 Computation times for one sample, averaged for each model group

ML model group	Time to calculate sound features (ms)	Time to calculate controller features (ms)	Average ML classification time (ms)	Total in-situ computation time per sample (ms)
All models	195.45	0.96	0.04	196.45
Base models	195.45	0.96	0.02	196.44
Ensemble models	195.45	0.96	0.06	196.47
Heterogeneous models	195.45	0.96	0.08	196.49
Homogeneous models	195.45	0.96	0.03	196.44

Due to its high sampling rate, the calculation of the sound features comprised a majority of the total computation time per ML sample. However, even using this sampling rate, the total computation time was under the 1 s maximum for this ML implementation. Although the classification time, which is the only component that is dependent on the ML algorithm selection, comprised a small amount of the total, comparisons may still be made. On average, the ensemble models required increased classification time than the individual models, which is expected due to their increased complexity. However, there was some overlap as, for example, the kNN model required more time than each of the ensembles except the EHV model. This is possible due to their use of decision tree base models, which have much shorter classification times than the other individual models studied.

4 Conclusions

The present study provided insight into how ML model configuration affects TCM performance in terms of classification accuracy, robustness to unbalanced datasets, and computation time. Overall, the ensemble models performed better than the individual models, with higher accuracies across a set of generalizability tests, lower accuracy standard deviations, higher 10-fold CV and LOGO-CV accuracies, and superior robustness to unbalanced class sizes. On the other hand, the individual models showed lower classification times, but only slightly lower overall processing times for real-time use.

Between the two types of ensemble models studied, the homogeneous ensembles performed better than the heterogeneous ensembles across most metrics. On average, they achieved higher 10-fold CV and LOGO-CV accuracies, lower classification times, stronger robustness to unbalanced datasets, higher precisions, and higher macro-averaged recall and F1 scores. However, the heterogeneous ensemble group generally achieved higher weighted recall and F1 scores. This improved understanding of ensemble ML model performance for TCM is valuable at a time when ensemble ML techniques have only just begun to be studied for this application.

This research raises a few additional research questions. For example, it would be interesting to investigate how the performance of a variety of heterogeneous and homogeneous ensemble models would compare to each other when the same high number of base models was used. It would also be valuable to study a wider variety of homogeneous ensembles, including ones using base models other than decision trees, and to compare their performance

to those of random forests and extra-trees models. Finally, for TCM systems to reach the generalizability levels necessary for real-world adoption, research will need to continue to focus on datasets generated from a variety of machining and environmental conditions.

Acknowledgment

This work was partially supported by the U.S. Department of Energy (DOE) through award DE-EE0008303, and by a Consortium for Enabling Technologies and Innovation fellowship awarded to Alexandra Schueller.

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The datasets generated and supporting the findings of this article are obtainable from the corresponding author upon reasonable request.

Nomenclature

ANN = artificial neural network
CV = cross validation
DT = decision tree
EHV = ensemble hard voting
ESV = ensemble soft voting
ET = extra-trees
kNN = k nearest neighbors
LOGO-CV = leave-one-group-out cross validation
ML = machine learning
RF = random forest
RFECV = recursive feature elimination using cross validation
RMS = root mean square
SD = standard deviation
SVM = support vector machine
TCM = tool condition monitoring
VB = flank wear

References

[1] Weller, E. J., Schrier, H. M., and Weichbrodt, B., 1969, "What Sound Can Be Expected From a Worn Tool?," J. Eng. Ind., **91**(3), pp. 525–534.
[2] Goebel, K., 1996, "Management of Uncertainty in Sensor Validation, Sensor Fusion, and Diagnosis of Mechanical Systems Using Soft Computing Techniques," Ph.D. dissertation., University of California, Berkeley, VA.
[3] Li, W., Guo, Y. B., Barkey, M. E., Guo, C., and Liu, Z. Q., 2011, "Surface Integrity and Fatigue Strength of Hard Milled Surfaces," International Manufacturing Science and Engineering Conference, Corvallis, OR, June 13–17, ASME Paper No. MSEC2011-50282, pp. 199–206.
[4] Denkena, B., Tonshoff, H. K., Friemuth, T., Mueller, C., Zenner, H., Renner, F., and Koehler, M., 2002, "Fatigue Strength of Hard Turned Components," Proceedings of 1st International Conference Manufacturing Engineering, Sani, Halkidiki, Greece, Oct. 3–4.
[5] Ghasempoor, A., Moore, T. N., and Jeswiet, J., 1998, "On-Line Wear Estimation Using Neural Networks," Proc. Inst. Mech. Eng.: J. Eng. Manuf., Part B, **212**(2), pp. 105–112.
[6] Ambhore, N., Kamble, D., Chinchanikar, S., and Wayal, V., 2015, "Tool Condition Monitoring System: A Review," Mater. Today: Proc., **2**(3–4), pp. 3419–3428.
[7] Binsaeid, S., Asfour, S., Cho, S., and Onar, A., 2009, "Machine Ensemble Approach for Simultaneous Detection of Transient and Gradual Abnormalities in End Milling Using Multisensor Fusion," J. Mater. Proc. Technol., **209**(10), pp. 4728–4738.
[8] Rangwala, S. S., and Dornfeld, D. A., 1989, "Learning and Optimization of Machning Operations Using Computing Abilities of Neural Networks," IEEE Trans. Syst. Man Cybern., **19**(2), pp. 299–314.
[9] Scheffer, C., and Heyns, P. S., 2004, "An Industrial Tool Wear Monitoring System for Interrupted Turning. Mechanical Systems and Signal Processing," Mech. Syst. Signal Proc., **18**(5), pp. 1219–1242.
[10] Dan, L., and Mathew, J., 1990, "Tool Wear and Failure Monitoring Techniques for Turning– A Review," Int. J. Mach. Tools Manuf., **30**(4), pp. 579–598.
[11] Roth, J. T., Djurdjanovic, D., Yang, X., Mears, L., and Kurfess, T., 2010, "Quality and Inspection of Machining Operations: Tool Condition Monitoring," ASME J. Manuf. Sci. Eng., **132**(4), p. 041015.
[12] Jemielniak, K., 1999, "Commercial Tool Condition Monitoring Systems," Int. J. Adv. Manuf. Technol., **15**(10), pp. 711–721.
[13] Kuntoglu, M., Aslan, A., Pimenov, D. Y., Usca, U. A., Salur, E., Gupta, M. K., Mikolajczyk, T., Giasin, K., Kaplonek, W., and Sharma, S., 2021, "A Review of Indirect Tool Condition Monitoring Systems and Decision-Making Methods in Turning: Critical Analysis and Trends," Sensors, **21**(1), p. 108.
[14] Ghosh, N., Ravi, Y. B., Patra, A., Mukhopadhyay, S., Paul, S., Mohanty, A. R., and Chattopadhyay, A. B., 2007, "Estimation of Tool Wear During CNC Milling Using Neural Network-Based Sensor Fusion," Mech. Syst. Signal Process., **21**(1), pp. 466–479.
[15] Kannatey-Asibu, E., Yum, J., and Kim, T. H., 2017, "Monitoring Tool Wear Using Classifier Fusion," Mech. Syst. Signal Process., **85**, pp. 651–661.
[16] Wang, J., Xie, J., Zhao, R., Zhang, L., and Duan, L., 2016, "Multisensory Fusion Based Virtual Tool Wear Sensing for Ubiquitous Manufacturing," Rob. Comput.-Integr. Manuf., **45**(4), pp. 47–58.
[17] Sick, B., 2002, "Online and Indirect Tool Wear Monitoring in Turning With Artificial Neural Networks: A Review of More Than a Decade of Research," Mech. Syst. Signal Process., **16**(4), pp. 487–546.
[18] Huang, Z., Zhu, J., Lei, J., Li, X., and Tian, F., 2020, "Tool Wear Predicting Based on Multi-Domain Feature Fusion by Deep Convolutional Neural Network in Milling Operations," J. Intell. Manuf., **31**(4), pp. 953–966.
[19] Kothuru, A., 2017, "Application of Audible Signals in Tool Condition Monitoring Using Machine Learning Techniques," Master's thesis, Rochester Institute of Technology, Rochester, NY. https://scholarworks.rit.edu/cgi/viewcontent.cgi?article=10890&context=theses
[20] Kothuru, A., Nooka, S. P., and Liu, R., 2018, "Application of Audible Sound Signals for Tool Wear Monitoring Using Machine Learning Techniques in End Milling," Int. J. Adv. Manuf. Technol., **95**(9–12), pp. 3797–3808.
[21] Wang, G., Yang, Y., and Li, Z., 2014, "Force Sensor Based Tool Condition Monitoring Using a Heterogeneous Ensemble Learning Model," Sensors, **14**(11), pp. 21588–21602.
[22] Yuan, J., Liu, L., Yang, Z., and Zhang, Y., 2020, "Tool Wear Condition Monitoring by Combining Variational Mode Decomposition and Ensemble Learning," Sensors, **20**(21), p. 6113.
[23] Riego, V., Castejon-Limas, M., Sanchez-Gonzalez, L., Fernandez-Robles, L., Perez, H., Diez-Gonzalez, J., and Guerrero-Higueras, A.-M., 2020, "Strong Classification System for Wear Identification on Milling Processes Using Computer Vision and Ensemble Learning," Neurocomputing, **456**(1), pp. 678–684.
[24] Hui, Y., Mei, X., Jiang, G., Tao, T., Pei, C., and Ma, Z., 2019, "Milling Tool Wear State Recognition by Vibration Signal Using a Stacked Generalization Ensemble Model," Shock Vib., **2019**(3), pp. 1–16.
[25] Javed, K., Gouriveau, R., Li, X., and Zerhouni, N., 2018, "Tool Wear Monitoring and Prognostics Challenges: A Comparison of Connectionist Methods Toward an Adaptive Ensemble Model," J. Intell. Manuf., **29**(8), pp. 1873–1890.
[26] ISO8688-2, 1989, *Tool Life Testing in Milling– Part 2: End Milling*, The International Organization for Standardization (ISO), Geneva, Switzerland, pp. 1–26.
[27] Tobon-Mejia, D. A., Medjaher, K., and Zerhouni, N., 2012, "CNC Machine Tool's Wear Diagnostic and Prognostic by Using Dynamic Bayesian Networks," Mech. Syst. Signal Process., **28**, pp. 167–182.
[28] Zhou, Y., Sun, B., Sun, W., and Lei, Z., 2020, "Tool Wear Condition Monitoring Based on a Two-Layer Angle Kernel Extreme Learning Machine Using Sound Sensor for Milling Process," J. Intell. Manuf., **33**(1), pp. 247–258
[29] Hsieh, W.-H., Lu, M.-C., and Chiou, S.-J., 2012, "Application of Backpropogation Neural Network for Spindle Vibration-Based Tool Wear Monitoring in Micro-Milling," Int. J. Adv. Manuf. Technol., **61**(1–4), pp. 53–61.
[30] Wu, Y., Hong, G.-S., and Wong, W. S., 2014, "Prognosis of the Probability of Failure in Tool Condition Monitoring Application—A Time Series Based Approach," Int. J. Adv. Manuf. Technol., **76**(1–4), pp. 513–521.
[31] Li, Z., Liu, R., and Wu, D., 2019, "Data-Driven Smart Manufacturing: Tool Wear Monitoring With Audio Signals and Machine Learning," J. Manuf. Process., **48**(4), pp. 66–76.
[32] Kothuru, A., Nooka, S. P., and Liu, R., 2017, "Cutting Process Monitoring System Using Audible Sound Signals and Machine Learning Techniques: An Application to End Milling," International Manufacturing Science and Engineering Conference, Los Angeles, CA, June 4–8, ASME Paper No. MSEC2017-3069.
[33] Lei, Y., He, Z., Ze, Y., and Chen, X., 2008, "New Clustering Algorithm-Based Fault Diagnosis Using Compensation Distance Evaluation Technique," Mech. Syst. Signal Process., **22**(2), pp. 419–435.
[34] Lee, L. C., 1986, "A Study of Noise Emission for Tool Failure Prediction," Int. J. Mach. Tool. Des. Res., **26**(2), pp. 205–215.
[35] Sadat, A. B., and Raman, S., 1987, "Detection of Tool Flank Wear Using Acoustic Signature Analysis," Wear, **115**(3), pp. 265–272.
[36] Schueller, A., 2021, "Ensemble Machine Learning Model Generalizability and Its Application to Indirect Tool Condition Monitoring," Master's thesis, Georgia Institute of Technology, Atlanta, GA.
[37] Shurrab, S., Almshnanah, A., and Duwairi, R. M., 2021, "Tool Wear Prediction in Computer Numerical Control Milling Operations Via Machine Learning," International Conference on Information and Communication Systems (ICICS), Valencia, Spain, May 24–26.

[38] Castejón-Limas, M., Sánchez-González, L., Díez-González, J., Fernández-Robles, L., Riego, V., and Pérez, H., 2019, "Texture Descriptors for Automatic Estimation of Workpiece Quality in Milling," Hybrid Artif. Intell. Syst., **11**, pp. 734–744.
[39] Elangovan, M., Devasenapati, S. B., Sakthivel, N. R., and Ramachandran, K. L., 2011, "Evaluation of Expert System for Condition Monitoring of a Single Point Cutting Tool Using Principle Component Analysis and Decision Tree Algorithm," Expert Syst. Appl., **38**(4), pp. 4450–4459.
[40] Jegorowa, A., Górski, J., Kurek, J., and Kruk, M., 2020, "Use of Nearest Neighbors (k-NN) Algorithm in Tool Condition Identification in the Case of Drilling in Melamine Faced Particleboard," Maderas: Cienc. Tecnol, **22**(2), pp. 189–196.
[41] Mannan, M. A., Kassim, A. A., and Jing, M., 2000, "Application of Image and Sound Analysis Techniques to Monitor the Condition of Cutting Tools," Pattern Recognit. Lett., **21**(11), pp. 969–979.
[42] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., et al., 2011, "Scikit-Learn: Machine Learning in Python," J. Mach. Learn Res., **12**(85), pp. 2825–2830.
[43] Wang, H., Yang, Y., Wang, H., and Chen, D., 2013, "Soft-Voting ClusteringEnsemble," *Multiple Classifier Systems. MCS 2013. Lecture Notes in Computer Science*, Z. H. Zhou, F. Roli, and J. Kittler, eds., Springer, Berlin/Heidelberg, pp. 307–318.
[44] Saqlain, M., Jargalsaikhan, B., and Lee, J. Y., 2019, "A Voting Ensemble Classifier for Wafer Map Defect Patterns Identification in Semiconductor Manufacturing," IEEE Trans. Semicond. Manuf., **32**(2), pp. 171–182.
[45] Saeed, U., Jan, S. U., Lee, Y.-D., and Koo, I., 2021, "Fault Diagnosis Based on Extremely Randomized Trees in Wireless Sensor Networks," Reliab. Eng. Syst. Saf., **205**(3), p. 107284.
[46] Zhang, B., 2020, "Improved Extremely Randomized Trees Model for Fault Diagnosis of Wind Turbine," Int. J. Sci., **7**(12), pp. 74–87.