

# Comparison of cutting tool wear classification performance with artificial intelligence techniques

COLANTONIO Lorenzo<sup>1,a\*</sup>, EQUETER Lucas<sup>1,b</sup>, DEHOMBREUX Pierre<sup>1,c</sup>  
and DUCOBU François<sup>1,d</sup>

<sup>1</sup>Machine Design and Production Engineering Lab, University of Mons, 7000 Mons, Belgium

<sup>a</sup>lorenzo.colantonio@umons.ac.be, <sup>b</sup>lucas.equeter@umons.ac.be,  
<sup>c</sup>pierre.dehombreux@umons.ac.be, <sup>d</sup>francois.ducobu@umons.ac.be

**Keywords:** Machining, Turning, Cutting Tool, Artificial Intelligence, Monitoring, Classification

**Abstract.** Optimal replacement of machining cutting tools is a major challenge in today's manufacturing industry. Due to the degradation of the tool during machining, late replacement of the tool leads to the risk of producing parts that do not meet technical specifications, while early replacement increases machine downtime and tool costs. To replace tools at the right time, it is necessary to monitor their degradation. Therefore, this paper compares the classification performance of different artificial intelligence approaches to classify the condition of cutting tools from cutting signals. Different approaches, namely: Artificial Neural Network (ANN), Support Vector Classifier (SVC), Random Forest (RF) and k-Nearest Neighbour (k-NN) are tested, and their performance is compared. It is highlighted that ANN and RF methods obtain better classification performances (88.8% and 86.4%, respectively) than the rest of the approaches (80%). Nevertheless, all approaches can monitor the degradation of cutting tools in a satisfactory manner (i.e., 80% accuracy). A comparison of training times highlights that training a neural network takes longer than the other approaches. However, with the computational power currently available, this is not an obstacle for their implementation in real applications as this training can still be achieved in a couple of minutes.

## Introduction

The condition of a cutting tool is of critical importance to the machined surface and the associated machining tolerances. A worn tool or a tool in an unsatisfactory condition does not allow the creation of machined surfaces of sufficient quality, which in consequence increases the cost of production [1]. Different tool replacement policies exist [2], but often tool replacement maintenance policies attempt to address this problem by replacing the tool well before its end of life which creates waste. This results in higher tool costs and increased machine downtime, further increasing production costs. As tool wear is an extremely complex and non-stationary phenomenon [3], the determination of the tool condition can be complex. Monitoring the degradation of the tool is thus necessary.

There are two types of monitoring: direct and indirect. The direct approach consists of measuring tool wear directly on the tool, but this requires the machining process to be stopped and results in increased machine downtime [4]. The indirect approach consists of measuring signals during the machining process to try to predict the state of the tool [5]. This has the advantage of allowing a continuous machining process with the least intrusive sensor installation possible. A review of the type of signals that can be recovered during machining is available in [6]. Several indirect monitoring methods exist, but lately, artificial intelligence methods are predominant as they can learn from machining data and adjust their prediction on a case-by-case basis. A systematic literature review describing all approaches present in the literature shows that there are mainly two approaches with AI: classification and regression [7]. Classification aims to monitor

the state of the tool via discrete values listing the state of the tool. Regression aims to follow the evolution of the tool by directly monitoring the wear. In some applications, it is not necessary to know exactly how the tool wear is evolving, so classification methods are used because of their ease of understanding.

In the literature, there are only a few comparisons of performance between different classification approaches. These comparisons are often made to highlight the particularity of one model compared to another, but a more general comparison is almost never made. This paper, therefore, proposes to compare the performance of some common artificial intelligence methods, namely: Artificial Neural Network (ANN), Random Forest (RF), Support Vector Classifier (SVC) and k-Nearest Neighbour (k-NN) implemented on the same database. The choice of these approaches is based on their ease of implementation and their ability to perform a classification for this application. All approaches are optimised and tested on identical data that homogeneously represents the different degradation states of the tool. The comparison of the quality of the results and the efficiency of the approach is realised.

### Methodology

To compare the performance of the different AI approaches, the database and the experimental conditions are described. Signals from the database correlated with tool wear are identified and used as input for the different AI approaches. The database is then divided into classes and a split between training and test data is made. Finally, all approaches, namely: ANN, RF, SVC and k-NN, are presented and optimised, and their results are highlighted. A comparison of the results is then made.

### Experimental Setup and Database

The database comes from experimental tests carried out on a CNC lathe (Weiler E35), which ensures a constant cutting speed throughout the machining process (Fig. 1). It is used to machine (longitudinal turning operation), C45 steel bars at variable cutting speeds with a CNMG120404-MF3 TP40 tool from SECO. This tool is one of the lower-grade tools to limit the amount of material wasted during testing. Table 1 shows the different cutting conditions used during the tests, only variations in cutting speeds are considered. The machine is instrumented with a force sensor (Kistler 9257B) that collect the cutting forces ( $F_i$ ) and torques ( $M_i$ ) during the machining operation. The sensor is mounted at the base of the tool and in the machine frame of reference,  $F_x$  corresponds to the feed force,  $F_y$  is the radial force and  $F_z$  is the cutting force. This sensor is mounted for indirect monitoring, i.e., to be as minimally intrusive as possible. Wear and cutting forces are measured every 2.8 minutes (corresponding to one piece). Wear is assessed according to ISO 3685 [8], which defines wear as the value of  $V_b$  (Fig. 2) measured directly on the tool using a microscope (Byameyee EU-1000X 3).  $V_b$  is measured as the size of the wear in zone B. This B area is located between the corner radius on one side and 1/4 of the wear area (area C where is located the notch wear) on the other side (Fig. 2)

A total of 30 tools are used to create the database, with the degradation of each tool being measured 6.4 times on average during its lifetime. The database therefore consists of 192 data points evenly distributed over the tool degradation. The measured cutting force corresponds to a 20 s signal sampled at 10 kHz and is processed to recover statistical and frequency values. The statistical analysis corresponds to the calculation of the average, the RMS value and the frequency analysis identifies the frequency and the maximum amplitude of the power spectral density.

To identify the relationships between the signals measured during machining and tool degradation, a Spearman correlation analysis is used on all signals. This correlation analysis is adapted to the size and non-normality of the data. The correlation analysis shows that the features most correlated with wear are the following:  $M_z$  RMS (correlation indicator: 0.89),  $F_x$  RMS (0.87), machining duration (0.84), chip length (total length machined) (0.84) and  $F_z$  RMS (0.79).

As cutting forces are strongly correlated with tool wear, they will be used in the following as inputs for the AI methods. It should be noted that the machining time and the chip length have the same correlation score as they are both dependent. The chip length is also dependent on the cutting speed. In this case, since only variations in the machining speed are taken into account, this indicator allows the method to indicate the change in cutting conditions.

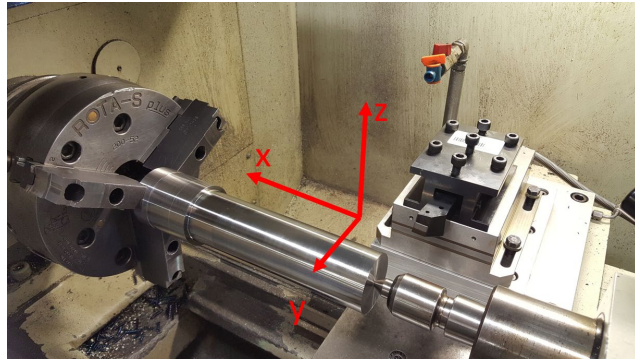


Fig. 1. Experimental turning setup, the tool is mounted into the cutting force and torque sensor.

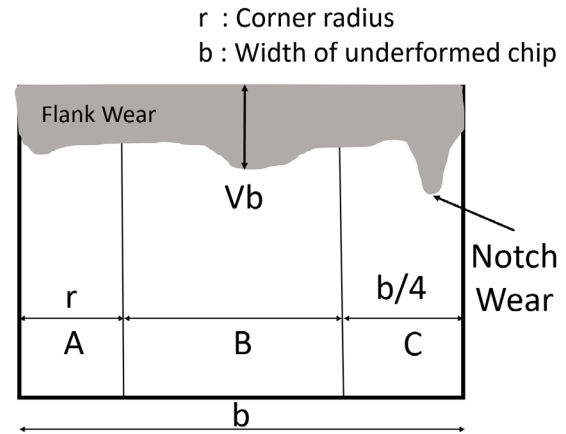


Fig. 2. Flank wear degradation  $V_b$  on the flank face.

Table 1. Experimental Cutting Condition.

Tool n°	Feed [mm/rev]	Cutting Speed [m/min]	Depth of cut [mm]
1 to 10	0.2	260	1
11 to 15	0.2	250	1
16	0.2	240	1
17 to 20	0.2	265	1
21 to 30	0.2	Variable: 240 to 260 (for each tool)	1

### Features Preparation for Tool Wear Classification

In a classification problem, it is necessary to define classes whose purpose is to define the different possible states of the tool. The degradation of a cutting tool is divided into 3 successive phases (Fig. 3): the first phase corresponds to the beginning of the tool's life, it shows little wear but degrades rapidly. This phase is generally short in relation to the life of the tool. The second phase is the longest and consists of a quasi-linear degradation of the tool, it is a regime phase in which the tool will spend most of the time. Finally, the third phase is the end of the tool's life, which can last more or less time depending on the cutting conditions.

Based on the ISO 3685 standard, it is often accepted that the end-of-life criterion for a cutting tool is when the size of the flank wear reaches 300  $\mu\text{m}$  [8]. This flank wear is called  $V_b$  and is presented in Fig. 2. It is therefore proposed that class 1 corresponds to wear between 0 and 150  $\mu\text{m}$ , class 2 corresponds to a wear between 150 and 300  $\mu\text{m}$  and class 3 corresponds to the end of the tool's life with wear exceeding 300  $\mu\text{m}$ . These values (150 and 300 microns) were chosen as they are generally used in the literature as end-of-life criteria.

The database is not uniform across all these classes, indeed, there are significantly more points in class 1 than in class 3 for example. To increase the number of points in each class, data augmentation is performed. This data augmentation consists of linearly interpolating 2 points of

the tool degradation and calculating an intermediate measurement point. This simple type of interpolation is sufficient given the number of measurements points in a complete trajectory; the interpolation error is low. This increases the number of possible points for training the AI but does not change the distribution of points in the different classes (73 % in class 1, 19% in class 2 and 8% in class 3). Nevertheless, with more data points, there will be more points to propose for training AI methods, which generally allows for faster convergence [9].

Artificial intelligence methods need training data to learn the relationships between the data and test data to verify that the model has learned correctly. It was chosen to select 15 points randomly per class to create the test database. For class 3, 15 data points correspond to 40% of the data in this class. This value is quite high (usually 25% of the data is used in testing) but this value does not impact the results presented in the following. By ensuring that each class contains the same number of data points, this ensures that an error in one class has the same overall importance regardless of the class.

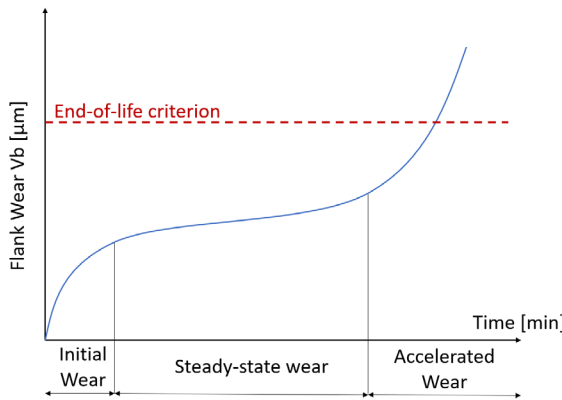


Fig. 3. Typical tool degradation for a given cutting condition.

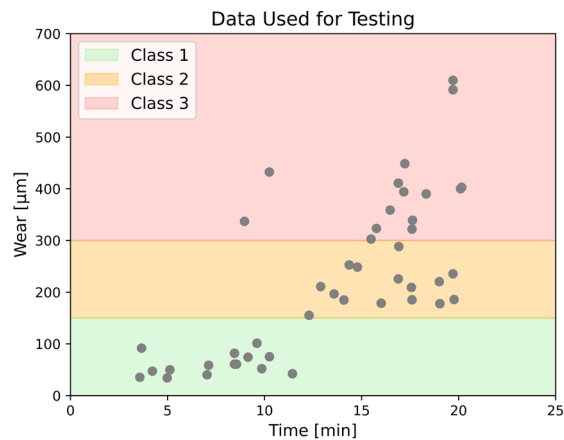


Fig. 4. Testing data randomly selected and their repartition in different classes.

### Artificial Neural Network

Neural networks are certainly the most popular intelligence method. Inspired by the way the brain works, this AI can represent highly non-linear relationships between input and output and is extensively represented in the literature [10]. There are multiple hyperparameters that influence the quality of the results, the most common are: the network architecture, the activation function used in each layer, the batch size, and the number of epochs. The most important hyperparameter is the network architecture which will create the relationship between the input and output of the network and is often chosen by trial and error. Table 2 shows the different hyperparameters used to obtain the best classification results. These and all other parameters presented in this paper are optimised by testing the influence of each parameter individually. In this case, for example, different architectures have been tested but the one chosen is the one that gives the best results. The network architecture is presented in Fig. 5.

The results are presented Fig. 6, the overall accuracy is 88.88%. Class 1 is always correctly classified but classes 2 and 3 present an accuracy of 86.7% and 80% respectively.

Table 2. Optimized hyperparameters for ANN.

Hyperparameter	Value
Algorithm	TensorFlow (Python) [11]
Architecture	2 hidden layers each containing 8 neurons
Activation function	1 <sup>st</sup> layer: Tanh, 2 <sup>nd</sup> layer: ReLu, output: Softmax
Batch size	5
Epoch	1000

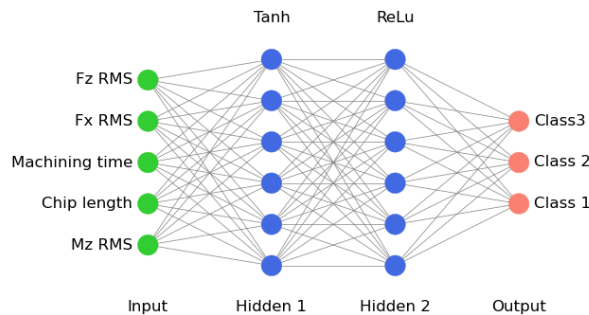


Fig. 5. Best Artificial Neural Network architecture.

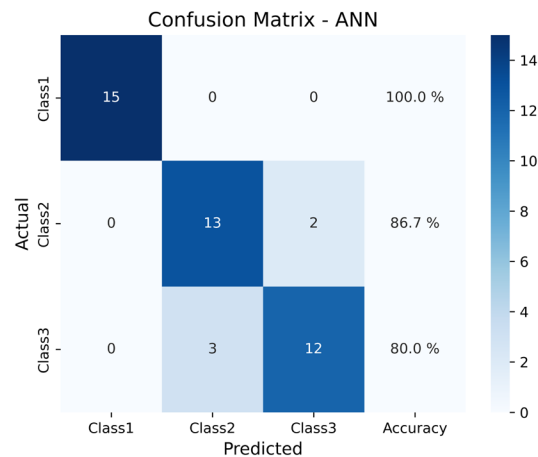


Fig. 6. ANN Confusion matrix – Overall accuracy : 88.8 %.

### Random Forest

The random forest classifier is a combination of classical tree classifiers. The Random Forest classifier consists at producing a set of tree classifiers (i.e., Number of estimator) to create an ensemble of classifiers, called forest [12]. To classify a state, each tree of the forest is interrogated, and the most given class is the predicted class (Fig. 7). Combining the results has the advantage of being more accurate than if each tree were taken individually, as the probability of a tree being wrong is higher than the probability of most trees being wrong.

Table 3 shows the different hyperparameters used to create the classifier. The number of estimators corresponds to the number of trees in the forest. The variation of this value can change the accuracy of the approach. The number of 500 is chosen as more estimators do not improve the overall accuracy. The maximum complexity of the trees is controlled by the “maximum depth”, the variation of this parameter slightly improves the results, a value too low leads to bad accuracy while a high value does not improve the results. To measure the quality of a split, the Gini impurity criteria is used [13].

The results obtained with the combination of hyperparameters previously identified are presented Fig. 8. The overall accuracy of the approach is 86.6%. Performance is consistent across all classes 2 and 3 with 80% accuracy.

Table 3. Optimized hyperparameters for RF.

Hyperparameter	Value
Algorithm	Sci kit Learn (Python) [13]
Number of estimators	500
Maximum depth	6
Criterion	Gini

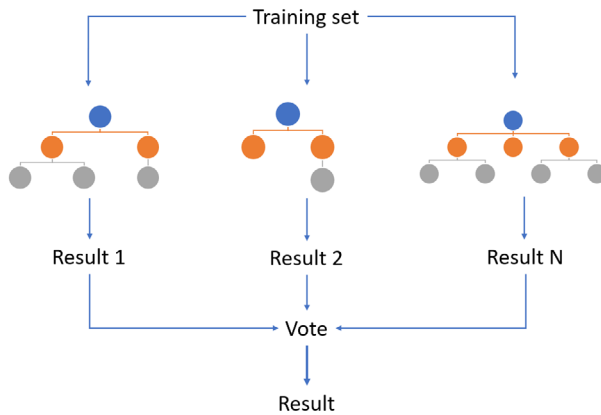


Fig. 7. Random Forest principle.

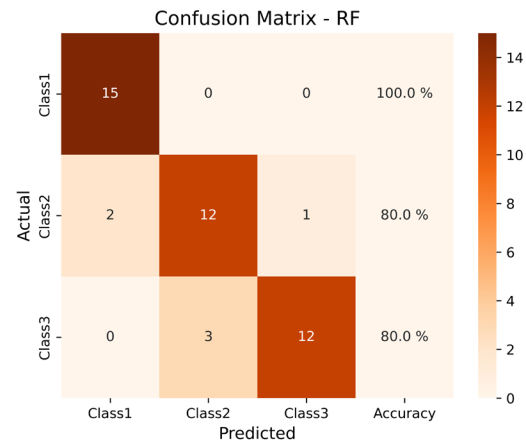


Fig. 8. Random Forest Confusion Matrix – Overall accuracy: 86.8%.

### Support Vector Classifier

Support vector machines, and more specifically the support vector classifiers, aim at finding an optimal hyperplane to separate different classes of data [14] (Fig. 9). In this application the kernel function that defines the different classes is the Radial Basis Function (RBF). This approach uses mainly 2 parameters: gamma and C. Gamma controls the curvature of the data separation and so controls the influence of samples selected by the model to be support vectors. The parameter C control the error rate by compromising between the correct classification against the maximization of the decision function. A compromise between these two parameters allows an efficient classification by controlling the shape of the classification area and the influence of outliers. Different combinations of parameters were tested but the combination of parameters that gives the best results is listed Table 4.

Fig. 10 shows the results obtained by the approach described above. The overall accuracy is 80%. Class 1 has no error, class 2 has the highest error rate with an accuracy of 66.7%, and class 3 has a performance of 73.3%.

Table 4. Optimized hyperparameters for SVC.

Hyperparameter	Value
Algorithm	Sci Kit Learn (Python) [13]
C	2
Gamma	1
Kernel	Radial Basis Function

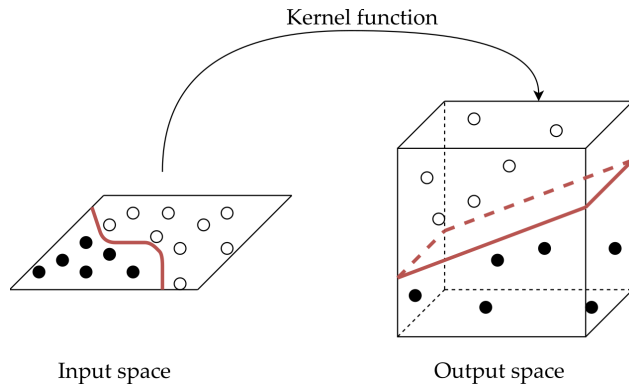


Fig. 9. SVC approach, under Creative Commons Attribution License [7].

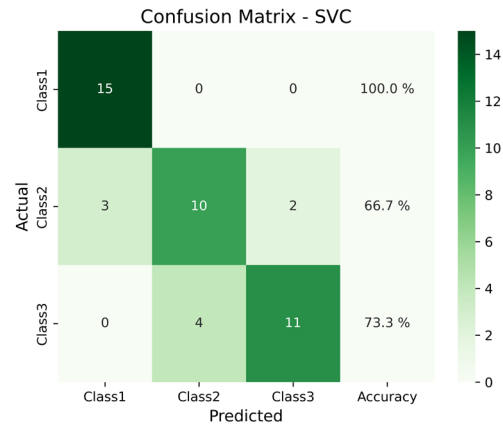


Fig. 10. SVC confusion matrix – Overall accuracy : 80%.

### k-Nearest Neighbour

K-Nearest Neighbour in classification uses the k nearest neighbour in the dataset reference of the input to determine the class of a new input [15]. The weight of each neighbour generally depends on the distance to the new inputs. Fig. 11 shows the example of the classification of a new element with a k value of 5. The 5 nearest data items and their distances are used to determine the class of the new item. Table 5 shows the most important parameters for this approach: the number of neighbour and the weight of the connection. In this approach, the best results were obtained with a number of neighbours of 8 and weight depending on the distance.

Fig. 12 presents the results. The overall accuracy is 80%. Only class 1 is correctly classified, class 2 and 3 have 66.7% and 73.3% accuracy respectively.

Table 5. Optimized hyperparameters for k-NN.

Hyperparameter	Value
Algorithm	Sci Kit Learn (Python) [13]
Number of neighbours	8
Weight	Distance
Algorithm	Auto

Example of k-NN with k = 5

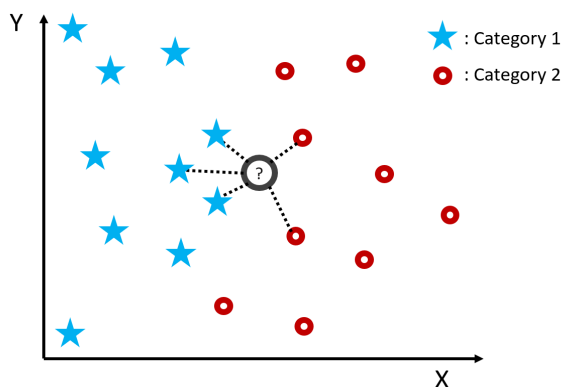


Fig. 11. k-NN approach with k = 5.

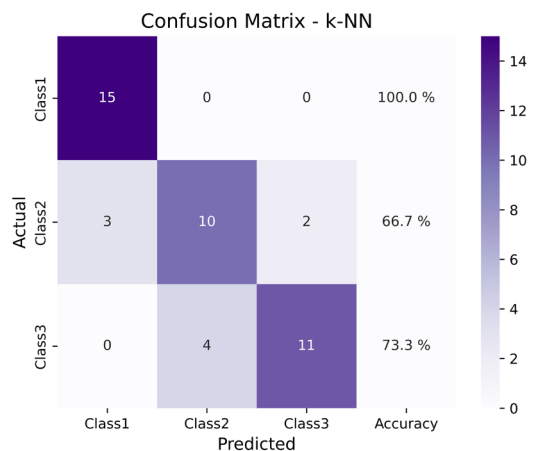


Fig. 12. k-NN confusion matrix – Overall accuracy: 80%

## Discussion

Table 6 compares the results obtained by the different approaches and the one with the highest overall score is the ANN method. It can be noted, however, that all approaches score well, with an average accuracy above 80%. Each method has its advantages and disadvantages, and the simple comparison of results does not allow to highlight them. For example, approaches such as k-Nearest Neighbour does not give good results if the dataset is not complete for all cutting conditions. On the other hand, approaches such as Neural Networks generally allow for a generalisation of the results when faced with never seen before cutting conditions.

*Table 6. Comparison of the overall accuracy, training and inference time for different AI approach.*

<b>Approach</b>	<b>Overall accuracy</b>	<b>Training Time</b>	<b>Inference Time</b>
ANN	88.8%	84.870s	0.080s
RF	86.6%	0.650s	0.050s
SVC	80%	<0.008s	<0.001s
k-NN	80%	<0.008s	<0.001s

An important point when using these approaches is also the computational time and resources required to obtain the results. All results presented in this paper are obtained on a single core of an Intel I7-9750H @ 2.6 GHz CPU. The different computational times, including training and validation are presented in Table 6. The training time considers the approach initialisation and computing time to fit the dataset. The inference time refers to the time needed by the approach to predict the classes presented in this paper. It is observed that the ANN approach has the longest training time compared to the other approaches. This is due to its training scheme. These considerations must be considered in relation to the available computational resources. However, it is important to note that the training only must be done once. With current computational resources, this is not a limitation as even ANN only takes less than 2 minutes to converge. It should also be noted that this interpretation is only valid for the amount of data in this database. For larger databases, the inference time of methods other than ANNs can increase considerably. In general, ANN methods have a longer training time but a relatively short inference time, which is not the case for other methods. Since training only needs to be done once, it is preferable to use methods with a constant low inference time such as ANN.

The position of the misclassification is also important. If the state of a tool is misclassified at the transition between two classes, this misinterpretation is less critical than if the tool is misclassified at middle of the class interval. Indeed, the misclassification reflects an error of some  $\mu\text{m}$  which does not significantly impact the quality of the machined surface. Fig. 13 shows the different positions of the wear incorrectly classified for each approach presented in this paper. From the figure, it appears that there are some data that are often misclassified in the middle of class 2. On this aspect, ANN has the lowest error than any other approaches. For each approach, the transition from class 2 to 3 leads to misclassification of the wear. As stated previously, this misclassification is not critical as this is only an error of less than 25  $\mu\text{m}$  in the estimation of  $V_b$ . SVC is the only approach that misclassified a very worn tool.



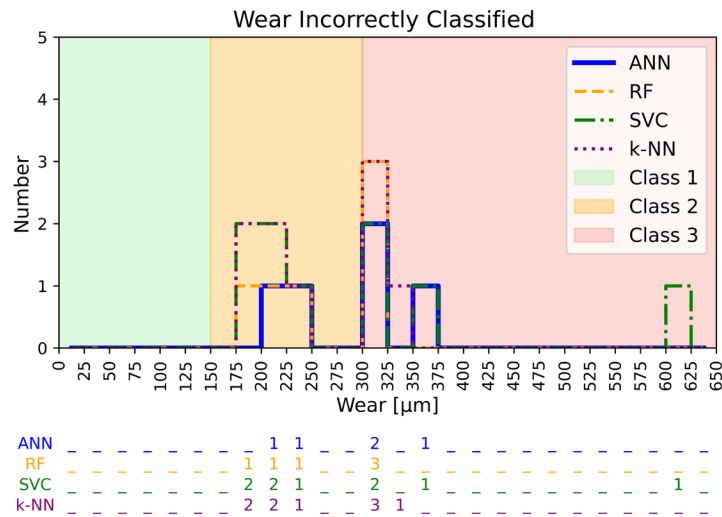


Fig. 13. Misclassified wear class for different techniques.

### Summary

Tool condition classification is an effective approach when the tool condition needs to be monitored. artificial intelligence methods are suitable for this purpose, but their performance can vary depending on the approach used. In this paper, a turning database from which cutting signals highly correlated with the wear are used with the following AI classification methods: ANN, RF, SVC, and k-NN. The tool condition is divided into 3 classes, each representing a phase of the tool life. The classes are defined based on the flank wear degradation of the tool ( $V_b$ ). Each AI approach is optimised to obtain the best achievable results and to be able to compare their relative performances. From the results, it appears that the ANN approach has the best accuracy with an overall accuracy of 88.8%. The second-best approach that achieves similar results is the RF with an accuracy of 86.6%. The other approaches have an average accuracy of 80%. Despite having the best accuracy, neural networks have by far the longest learning time (85s) compared to less than 1 s for all the other approaches. This learning time must be considered as with a larger and more complete database, the learning and inference time can greatly grow. Nonetheless, with the actual computing power, these considerations should not represent any obstacle to real applications as the training can be realised off-line. The results presented in this paper are limited to an ideal case to compare the performance of the different approaches presented. In industrial practice, it is necessary to consider the changes in cutting conditions as well as other variables that can influence the process. Nevertheless, the results of this paper allow a comparison of the approaches presented.

### Conflicts of interest

The authors declare that they have no conflicts of interest.

### References

- [1] W. Xu, L. Cao, Optimal tool replacement with product quality deterioration and random tool failure, *Int. J. Prod. Res.* 53 (2015) 1736-1745. <https://doi.org/10.1080/00207543.2014.957878>
- [2] A. Jeang, Reliable tool replacement policy for quality and cost, *European J. Operat. Res.* 108, (1998) 334-344. [https://doi.org/10.1016/S0377-2217\(96\)00368-2](https://doi.org/10.1016/S0377-2217(96)00368-2)
- [3] F. Klocke, *Manufacturing Processes 1*, RWTHedition, Springer, Berlin/Heidelberg, Germany, 2011.
- [4] D.E.D. Snr, Sensor signals for tool-wear monitoring in metal cutting operations—a review of methods, *Int. J. Mach. Tool. Manuf.* 40 (2000) 1073-1098. [https://doi.org/10.1016/S0890-6955\(99\)00122-4](https://doi.org/10.1016/S0890-6955(99)00122-4)

- [5] M. Kuntoğlu, A. Aslan, D.Y. Pimenov, Ü.A. Usca, E. Salur, M.K. Gupta, T. Mikolajczyk, K. Giasin, W. Kaplonek, S. Sharma, A review of indirect tool condition monitoring systems and decision-making methods in turning: Critical analysis and trends, *Sensors* 21 (2020) 108. <https://doi.org/10.3390/s21010108>
- [6] A. Siddhpura, R. Paurobally, A review of flank wear prediction methods for tool condition monitoring in a turning process, *Int. J. Adv. Manuf. Technol.* 65 (2013) 371-393. <https://doi.org/10.1007/s00170-012-4177-1>
- [7] L. Colantonio, L. Equeter, P. Dehombreux, F. Ducobu, A systematic literature review of cutting tool wear monitoring in turning by using artificial intelligence techniques, *Machines* 9 (2021) 351. <https://doi.org/10.3390/machines9120351>
- [8] ISO 3685—Tool Life Testing with Single-Point Turning Tools. 1993, Available online: <https://www.iso.org/fr/standard/9151.html> (last accessed 01/12/2022)
- [9] T. Zhao, Y. Liu, L. Neves, O. Woodford, M. Jiang, N. Shah, Data augmentation for graph neural networks, in *Proceedings of the AAAI Conference on Artificial Intelligence* 35 (2021) 11015-11023.
- [10] A. Abraham, *Artificial Neural Networks*, in: *Handbook of Measuring System Design*, John Wiley & Sons, Ltd.: Chichester, UK, 2005.
- [11] Tensorflow, Available online: <https://www.tensorflow.org/?hl=en> (accessed 01 December 22)
- [12] A. Parmar, R. Katariya, V. Patel, A review on random forest: An ensemble classifier, in: *International Conference on Intelligent Data Communication Technologies and Internet of Things*, Springer, Cham, 2018, pp. 758-763.
- [13] Sci-Kit Learn, Available online: <https://scikit-learn.org/stable/> (accessed 01 December 22)
- [14] I. Steinwart, A. Christmann, *Support vector machines*, Springer Science & Business Media, 2008.
- [15] L. Jiang, Z. Cai, D. Wang, S. Jiang, Survey of improving k-nearest-neighbor for classification, in: *Fourth international conference on fuzzy systems and knowledge discovery (FSKD 2007)*, 1 (2017) 679-683).