Comparison of artificial intelligence techniques for cutting tool condition monitoring

COLANTONIO Lorenzo[1,a *], EQUETER Lucas[1,b], DEHOMBREUX Pierre[1,c] and DUCOBU François[1,d]

[1]Machine Design and Production Engineering Lab, University of Mons, 7000 Mons, Belgium

[a]Lorenzo.colantonio@umons.ac.be, [b]lucas.equeter@umons.ac.be, [c]Pierre.dehombreux@umons.ac.be, [d]francois.ducobu@umons.ac.be

Keywords: Machining, Artificial Intelligence, Wear, Monitoring

Abstract. In manufacturing, tool wear monitoring is crucial as it directly influences production quality and operating costs. Inaccurate replacement strategies can result in increased costs and substandard parts. Various Artificial Intelligence (AI) methods have been proposed to monitor tool wear using cutting signals, but a comprehensive performance comparison is lacking. This paper evaluates three distinct AI approaches: Artificial Neural Networks (ANN), Support Vector Machines (SVM), and K-Nearest Neighbours (K-NN). The selection of these approach is based on their learning mechanisms. Each method is optimized using a GridSearch algorithm and their real-time wear monitoring capabilities are compared. The results shows that all AI techniques monitored tool wear with similar precision, making it challenging to draw a definitive conclusion in this regard. The choice of the most appropriate AI method is heavily dependent on the manufacturing environment. For large-scale manufacturing under similar cutting conditions, K-NN and SVM are a good choice. The ANN is better suited to all scenarios, but particularly where there are substantial fluctuations in cutting conditions or, in general, larger databases.

Introduction

The quality of a machined surface is related, among others, to the condition of the tool that was used to produce it. Despite often being overlooked in industrial context, the wear condition of the tool is of critical importance. Indeed, a worn cutting tool will lead to poor quality in terms of geometrical, residual constraints and tribological standpoints. In industrial context, tool replacement policies are often a compromise between the stopping of the operation and the quality of the production. Therefore, the choice of the replacement policies is of crucial importance as a wrong policy implies higher operating costs [1]. To optimize the replacement policies and by consequence limit the production of substandard parts, there is a need to monitor the degradation of the tool. The ISO 3685 standard defines the indicator to evaluate the state of the tool as the size of the flank wear (VB), thus most of monitoring methods focus on methods to estimate this type of wear.

The detection of the wear is often realised indirectly, by using signals from the cutting process to estimate the state of the tool. This approach has the advantage of not needing to stop the machining to inspect the state of the tool, but it needs to instrumentalize the machine. A complete analysis of the different signals used for indirect monitoring and their features is presented in [2]. Numerous strategies have been developed to assess the condition of the tool and therefore propose to follow its degradation path during machining. Earlier methods used statistical analysis or stochastic modelling of the process [3].

Recently and with industry 4.0, more Artificial Intelligence (AI) approaches tend to monitor the state of the tool. These approaches have the benefit of learning directly from the data and therefore automatically map the complex non-linearity between the condition monitoring signals and the state of the tool even in variable cutting conditions. Numerous applications, each employing distinct AI techniques, have been suggested in various studies [4]. Each application has its unique set of input signals, pre-processing methods, … which differ from one another, making it challenging to get a comprehensive comparison of performance between methods [5]. In general, the most common approach is to use an artificial neural network on cutting forces to monitor the degradation in real time [6].

Despite many approaches and generally good results obtained with AI [7], there is a lack of clear comparison of different AI methods. The only comparison is often limited to different pre-processing or data acquisition techniques or are limited to classification purpose [8]. In this paper, it is proposed to compare the performance of different AI regressors to monitor the state of the tool from cutting signals during machining. This article therefore compares 3 AI approaches, namely: Artificial Neural Networks (ANN), Support Vector Machine (SVM) and K-Nearest Neighbours (K-NN). The selection of these methods stems from their differences of concept to learn from the data. Each of them is explained in its respective section below.

The novelty of the approach consists of comparing the performance of different AI techniques using an identical optimisation and evaluation process. To optimise each approach, a GridSearch algorithm is used to identify the best combination of hyperparameters. The evaluation is realised on variable cutting conditions to ensure that the approaches can generalize their results on previously unseen cutting conditions. Differences in performance between the methods are assessed using a numerical indicator, and a visual representation of monitoring quality is also provided.

Description of Database

The database presented in this article is from experimental turning tests. The lathe is a Weiler E35 that is used to machine C45 steel bars (Figure 1). The selected cutting tool is the CNMG120404-MF3 TP40 tool from SECO and it is one of the lowest grades to favorize the apparition of wear and limit the test duration and the quantity of wasted materials. A total of 30 tools are used under different cutting conditions. Those cutting conditions are presented in Table 1 and are selected to observe the impact of variation of the cutting speed on the wear.

Table 1 Testing Cutting Conditions

Test n°	Cutting Speed [m/min]	Feed [mm/rev]	Depth of cut [mm]
1 to 10	260	0.2	1
11 to 15	250	0.2	1
16	240	0.2	1
17 to 20	265	0.2	1
21 to 30	Variable during life: 240 to 260	0.2	1

Table 2 Measured Cutting Parameters

Measured value	Equipment	Denomination
Cutting force	Kistler 9257B	Fx, Fy, Fz
Cutting torque	Kistler 9257B	Mx, My, Mz
Wear of the tool	Byameyee EU-1000X 3	VB

During the turning tests, multiple quantities are measured at a recurrent inspection interval of 2.8 minutes and are listed in Table 2. The cutting forces and torques are measured during machining. To inspect the state of the tool, the machining is stopped, and the wear is measured with a Byameyee EU-1000X 3 microscope according to the ISO 3685 standard. This standard defines VB_{max} and VB as shown in Figure 2 and limits the maximum amount of wear before the tool is considered worn. For the test and according to the standard, it is considered that a tool is worn if the value of VB reaches 300 μm.

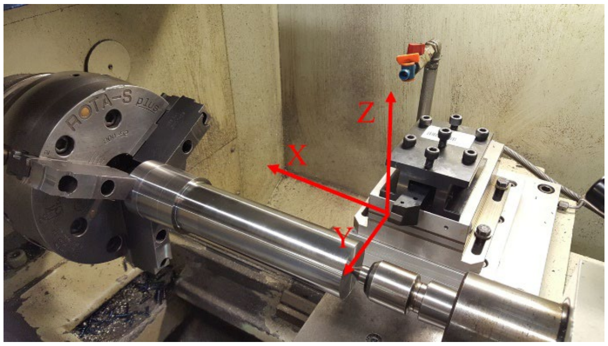

Figure 1 Experimental Set-Up

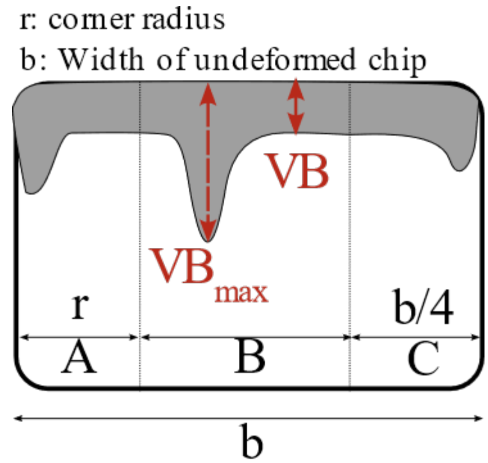

Figure 2 Flank Face Wear Definition

The cutting forces and torques are temporal signals that are pre-processed to extract meaningful information from it. There exists plenty of pre-processing approach but the most common have been selected based on their effectiveness in the literature [5]. The selected pre-processing techniques are: Mean, Root Mean Squared, Skewness, … In addition to these values, other indicators are considered such as the cutting time and the machine length.

These values are not all correlated with the wear. To estimate their correlation with it, a Spearman's correlation analysis is performed according to [9]. This correlation analysis is the most adapted to the complete dataset. The most correlated signals and their associated correlation are: Mz RMS (correlation indicator: 0.89), Fx RMS (0.87), machining duration (0.84), total length machined (0.84) and Fz RMS (0.79). These signals are therefore used as inputs to all the AI approaches presented in this study.

Methodology to Compare and Optimize the Results

In the following, multiple AI approaches are compared. Each approach can be tuned through different set of hyperparameters that can influence the quality of the results obtained with the approach. In this article, it is proposed that the determination of hyperparameters is systematically determined through a GridSearch optimization algorithm. This optimization approach consists of testing each combination of hyperparameters that are defined for each AI regressor.

To ensure that the approach does not overfit to the database, this GridSearch algorithm uses a 5-fold cross-validation to assess model performance. In this cross-validation, the dataset is split into five parts. Each part serves as a test set once, while the model is trained on the remaining data. The process is repeated five times, and the results are averaged to provide a robust performance estimate. The number of five folds is chosen to have 20% of the database for testing while training on the remaining 80%. This method allows to systematically identify the best combinations of hyperparameters for each approach and the 5-folds approach ensure that there is no overfitting to obtain the results presented in the following. The optimization is realised on Intel I7-9750H @ 2.6 GHz CPU.

Once the best hyperparameters are identified, a new optimized model is trained and tested. To compare the performance on an objective basis, the database is divided into a training database and a testing database. As it is common practice in AI, the training database accounts for 80 % of the whole database and the remaining 20 % are used for testing. The testing database consists of 6 trajectories that are: 23, 24, 26, 27, 28 and 29 (cf. Table 1). These trajectories are selected as they have variation in their cutting conditions between two inspections (Figure 3) and are therefore good indicator to assess the generalization capability of the AI approaches. Indeed, these variations are different from one another and from training database trajectories and therefore are not represented in the training database. This feature thus allows to compare the monitoring performance on previously unseen cutting conditions during the life of the tool.

The selected performance indicator for this study is the Mean Squared Error (MSE). It consists of computing the average of the squares of the n errors (e_i) between the n real value of VB and the n estimated one (Figure 4 and Eq.1). In the following, the MSE is computed for each testing trajectory and then the mean MSE is computed on the whole testing dataset. For the MSE, a lower value indicates lower error.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n} e_i^2 \quad (1)$$

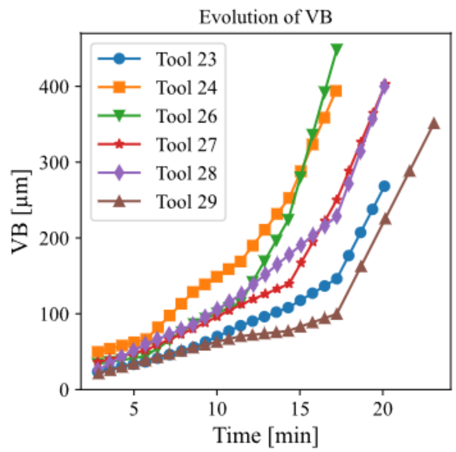

Figure 3 Testing Trajectories

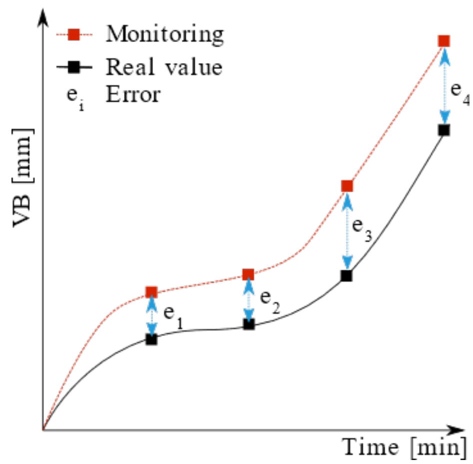

Figure 4 MSE Computation

Description of the AI Approaches

In this study, 3 AI approaches are compared: Artificial Neural Networks, Support Vector Machine and K-Nearest Neighbours.

Artificial Neural Networks (ANNs) are a leading approach in machine learning. Inspired by the structure and function of the human brain, they consist of interconnected layers of neurons that can model complex, non-linear relationships (Figure 5). The performance of an ANN is primarily dictated by its architecture. The neural network is constructed by layer, each one having a specific number of neurons with their respective activation function. The choice of structure is often induced by the complexity of the problem. Generally, more neurons and layers can model more complex function [10]. However, networks that are too deep may be less reliable if they are not warranted by the complexity of the problem.

The second approach is a Support Vector Machine (SVM). SVM in regression, known as Support Vector Regression (SVR), is a powerful regression tool. It operates by first applying a kernel function to the input space to transform the input data into a higher-dimensional space, enabling the conversion of a non-linear problem into a linear one (Figure 7). In addition to the kernel, there are 2 main hyperparameters: ε and C. ε defines the margin of tolerance where errors are not penalized, effectively controlling the width of the “tube” around the regression line. The C parameter, on the other hand, manages the trade-off between model complexity and its generalization ability [11].

The last approach is the K-Nearest Neighbour (K-NN) algorithm. It utilizes the structure of the input space of a given dataset to predict the value of a new entry. Specifically, it constructs a multidimensional feature space based on the input data and determines the value of a new entry based on the values of its ‘k’ nearest neighbours within this feature space. As shown Figure 6, when 'k' is set to 4, the algorithm evaluates the new data point by considering the four closest existing points in the input space. The predicted value of the new entry is then typically determined by an average or weighted average of the values of its 'k' nearest neighbours [11].

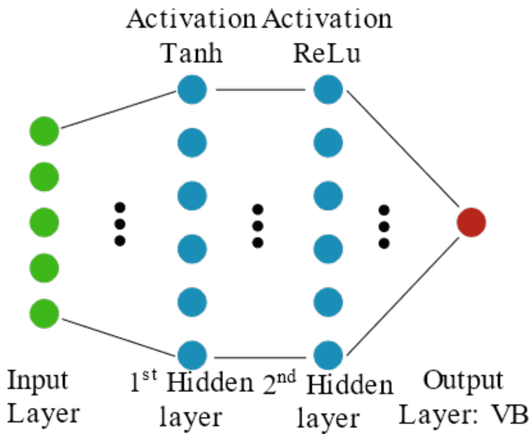

Figure 5 ANN Principle

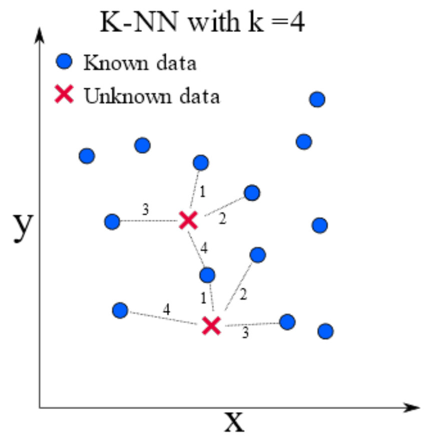

Figure 6 K-NN Principle

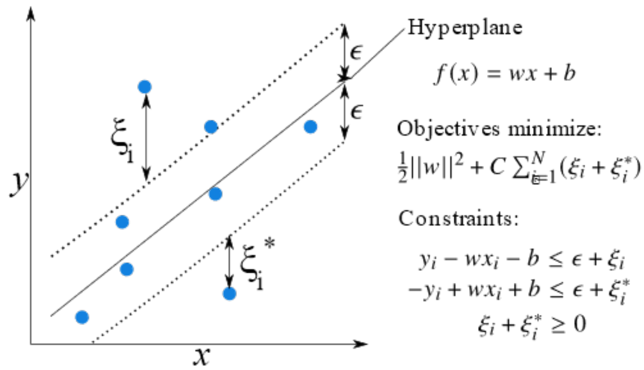

Figure 7 SVM Principle

Results – Artificial Neural Networks

To determine the optimal network architecture, a series of architectures are systematically evaluated. The optimization parameters of the GridSearch algorithm are the activation function (hyperbolic tangent – Tanh, a Rectified Linear Unit – Relu and sigmoid), the number of neurons (1 to 40) and the number of layers (1 to 20). A total of 1600 architectures have been tested. On average, the training time is around 90 seconds. Figure 8 shows the results of the optimization. The Figure 9 shows that an increase in the number of layers has a negative impact on the quality of the monitoring. Also, above a certain number of layers, there is no benefit to add neurons in the layer. The data shown in the figures suggest that the ideal network consists of 2 to 6 layers, each with 6 to 10 neurons. Further optimization led to a network that is composed of two layers, each with six neurons. The first layer uses a hyperbolic tangent (Tanh) activation function, while the second layer utilizes a Rectified Linear Unit (Relu) activation function (Figure 5).

Materials Research Proceedings 41 (2024) 1962-1971 https://doi.org/10.21741/9781644903131-217

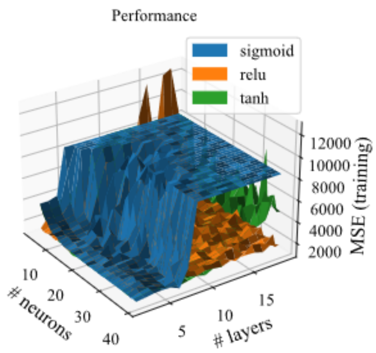

Figure 8 ANN - Optimization Results

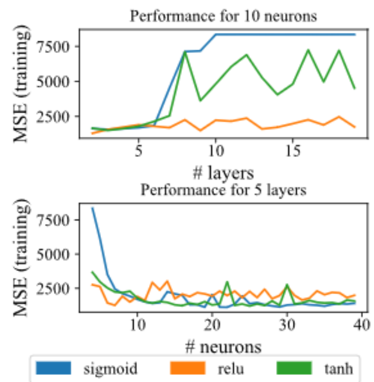

Figure 9 ANN – Optimization Detailed

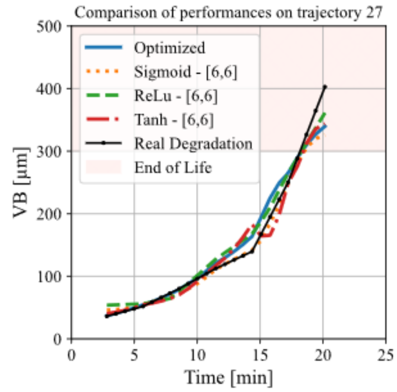

Figure 10 ANN - Choice Impact on Monitoring

Figure 10 shows the difference between the optimized network and other network with 2 layers with 6 neurons each that exclusively employs a "ReLu", "Tanh", or "Sigmoid" activation function. As all networks tend to converge to the same range of MSE, the observation of the best performance is not visible in Figure 10. On average, over the whole testing dataset, the optimized network has better performance than the other networks as presented Table 3.

Table 3 ANN - Comparison of Performance for the Testing Trajectories

Method	MSE 23	MSE 24	MSE 26	MSE 27	MSE 28	MSE 29	Average MSE	Average without 26
Optimized	190.5	193.9	2412.4	379.8	102.4	1082.5	726.9	389.8
ReLu only	414.4	292.2	2237.6	362.6	343.0	548.5	699.7	392.1
Tanh only	1060.5	976.9	567.9	284.5	186.1	3159	1039.2	1133.4
Sigmoid only	308.5	399.0	1392.67	387.76	343.8	911.6	623.9	470.2
Overall best	Optimized	Optimized	Tanh	Tanh	Optimized	ReLu	Optimized	Optimized

Results – Support Vector Machine

The combination of the kernel function, the parameter C and the parameter ε of the SVM approaches can be intricated. The tested GridSearch parameters and their values are the kernel function (Linear, Polynomial (1 to 10 degrees), Radial Basis Function (RBF), Sigmoid), the regulizer C (1 to 150) and ε (0.1 to 1 by 50 steps of 0.02). This choice of parameters space is made to ensure that a wide range of values are tested. A total of 97500 combinations of parameters are tested. On average, the training time is around 0.05 seconds.

The optimized parameters for each kernel and their respective global performances on the testing dataset are presented Table 4. Figure 11 shows the different performances of different kernels functions depending on the hyperparameters. This figure shows that the best performances are achieved with the RBF function and that there is a wide region where the results obtained are in the same range. It is also observable that the parameter ε has no effect on the results. On the other hand, the sigmoid kernel is not able to obtain good results and is not presented in the top of Figure 11.

To visually represent the impact of the choice of hyperparameters, Figure 12 shows the best combination of hyperparameter for each Kernel function. On this Figure, it is clearly observed that the sigmoid kernel function is to avoid as it tends to largely underestimate the wear at the end-of-life of the tool. The same phenomenon is observed for the linear approach. The polynomial approach tends to have a larger error in the middle of the life of the tool but at the end-of-life, there

Materials Research Proceedings 41 (2024) 1962-1971 https://doi.org/10.21741/9781644903131-217

is less difference than other models. Finally, the RBF kernel gives the best results and follows almost perfectly the real degradation.

Table 4 SVM – Best Hyperparameters for Each Kernel and their Performance on Testing Set

Kernel	*C*	*ε*	Global MSE (testing set)
Radial Basis Function	51	0.85	670
Linear	9	0.79	1352
Polynomial (degree: 3)	1	0.22	6096
Sigmoid	1	0.78	5167

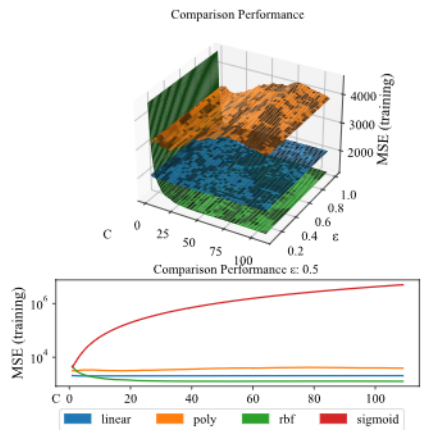

Figure 11 SVM – Optimization Results

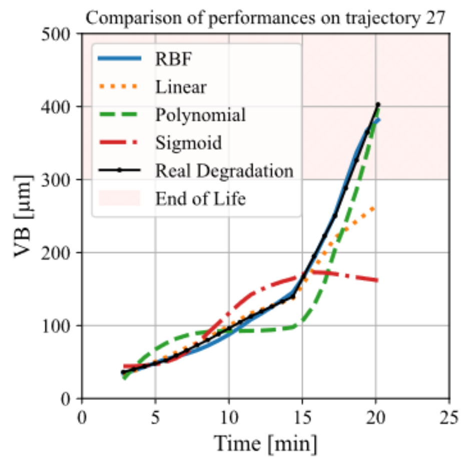

Figure 12 SVM – Kernel Impact on Performance

Results – K-Nearest Neighbour

Two primary hyperparameters significantly influence the results: the distance computation method and the number of neighbours. The distance computation can be either "uniform", where each neighbour is given equal weight, or "distance", where the weight of each neighbour is inversely proportional to its distance from the new point. The number of neighbours determines the size of the local neighbourhood used for the prediction. The optimization considers both distance computation and a neighbourhood from 1 to 150 neighbours. In total 300 combinations have been tested for an average fitting time of less than 0.002s for each.

The Figure 13 illustrates the training performance of K-NN, considering the weight function and the number of neighbours. The area labelled as the "Best performance zone" means that the MSE is around 10% of the optimal value. It is observed that the optimal zone is larger for the "distance" weight computation. Furthermore the "distance" weight computation has the best minimal MSE with a neighbourhood of 28. Given the broader optimal zone and the lower minimal MSE of the distance weight calculation, this method is recommended.

Figure 14 shows the difference of the best performance for "uniform" and "distance" approach. As their MSE are similar, the results are almost identical. There is simply a small difference at the end-of-life of the tool but this error as no impact in real life application.

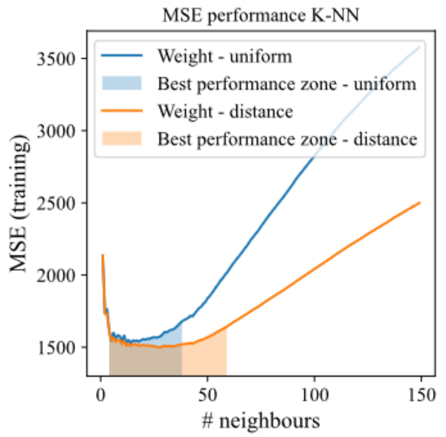

Figure 13 K-NN – Optimization Results

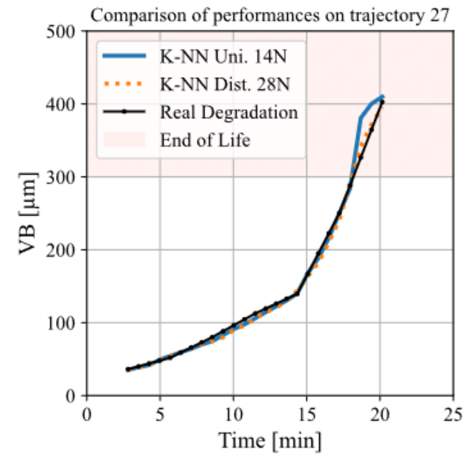

Figure 14 K-NN Hyperparameters Impact

Results – Comparison

As each AI model is optimized, it is possible to compare their performance against each other. Table 5 shows the MSE of each optimized approach on the testing set. It is observed that all the approaches can achieve similar performance on average. Table 5 also shows that even though neural networks does not have the best mean MSE, it is the method that presents with the best results over the testing trajectories. The worst MSE for ANN is due to the error on trajectory 26 which leads to a significant increase in the mean MSE.

Table 5 Comparison of Performances for the 3 Presented Approaches

Method	MSE 23	MSE 24	MSE 26	MSE 27	MSE 28	MSE 29	Average MSE
ANN	190	193	2412	379	102	1082	694
SVM	124	1002	1002	41	165	1790	670
K-NN	513	982	523	44	213	1449	613
Best	SVM	ANN	K-NN	SVM	ANN	ANN	ANN

Figure 15 visually represent the performance of the different approaches on the testing trajectories. In Trajectory 23, both ANN and SVM provide similar wear estimations, while K-NN tends to overestimate the wear, predicting VB above 300 µm. Trajectory 24 reveals comparable monitoring for SVM and K-NN, with ANN proving more reliable during the tool's lifespan, but less so post end-of-life. Trajectory 26 displays similar errors across all estimators, but ANN overestimates the tool's wear, resulting in a high MSE score. For Trajectory 27, SVM and K-NN estimations align closely and accurately track the wear, whereas ANN deviates. In Trajectory 28, all approaches are similar, but ANN is slightly more accurate. Lastly, Trajectory 29 shows a steep degradation after 17 minutes, causing errors in the estimations. All estimators predicted higher wear, decreasing towards the end, with K-NN even reducing the VB value below the end-of-life. The analyse of performance, does not allow to clearly identify a best regressor over the other. Each model has its own strengths and limitations, and the choice of the best model depends on various factors beyond just performance metrics.

The K-NN model's effectiveness is heavily sensitive to the quality of the database, especially the noise and outliers within it. Furthermore, the amount of data can also be a limitation. In the database presented in this article, the number of points is limited. However, in industrial practice, databases can contain thousands of data points and cutting conditions. Given that this method requires the computation of the distance between each point for monitoring, it can be challenging to scale in industrial practice. The limitation of SVM is also in the database. Although it is less sensitive to outliers compared to K-NN, it struggles with scalability when dealing with large datasets. When the dataset exceeds 10,000 data points, the scalability of SVM decreases, and the inference time significantly increases, rendering it unsuitable for real-time predictions in the cutting process. Despite these constraints, K-NN and SVM can be highly effective in scenarios with limited variations on the cutting conditions, such as the production of similar items in large series.

Lastly, ANN has the potential to overfit the data, which can lead to disastrous results in industrial applications. However, if implemented correctly, ANN is the most scalable among the three models. It can handle large datasets effectively, and its architecture can be adjusted to accommodate different types of signals and complexity. Despite having a significant higher training time than SVM and K-NN, ANN keeps a relatively low inference time independently to the size of the database. This adaptability makes ANN a viable option for a variety of use-cases.

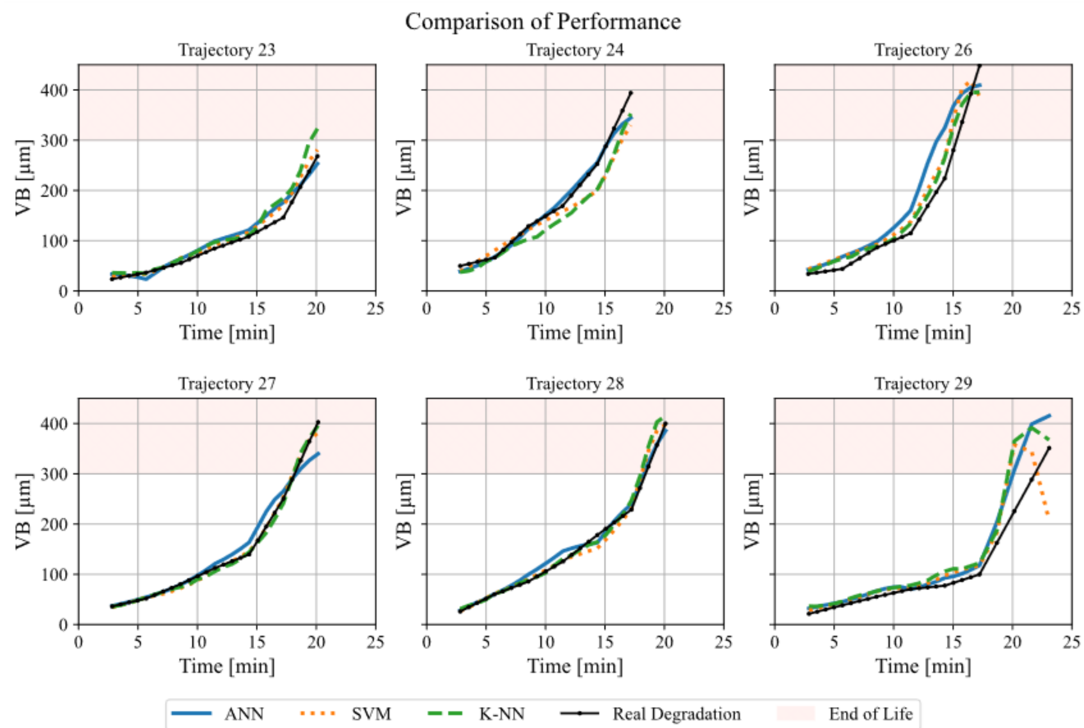

Figure 15 Comparison of Performance for each AI Approach

Conclusion

This article compares the performance of 3 different AI regressors, namely: ANN, SVM and K-NN. Each approach is optimised to find the best hyperparameters through a GridSearch algorithm and their performance are compared in terms of MSE. In this article and with the database presented above, the best ANN (MSE: 694) is a network with 2 hidden layers and 6 neurons in each of them. The first layer has a Tanh and the second a ReLu activation function. For SVM (MSE: 670), the best approach is reached with the RBF kernel, the parameter C is set to 51 and ε is 0.85. Finally, the K-NN (MSE: 613) approach best parameters are a "distance" weight computation and around 28 neighbours.

The performance comparison of each regressor on actual degradation trajectories shows similar results across all methods, as indicated by the similar MSEs. The best regressor for an industrial use depends on specific case factors. For instance, SVM and K-NN are negatively affected by the amount and quality of data, especially if it is noisy. On the other hand, ANN benefits from larger datasets and can be tailored to problem complexity. Finally, this article suggests that a good regressor for tool condition monitoring in big series industrial application are K-NN and SVM for their ease of implementation and optimisation. For more complex industrial cases with important variations in cutting conditions, machine, tools, and in general amount of data, ANN approaches appear to be more adaptable.

Conflicts of interest

The authors declare that they have no conflicts of interest.

References

[1] W. Xu and L. Cao, ‘Optimal tool replacement with product quality deterioration and random tool failure’, *Int. J. Prod. Res.*, vol. 53, no. 6, pp. 1736–1745, Mar. 2015. https://doi.org/10.1080/00207543.2014.957878

[2] M. Kuntoğlu *et al.*, ‘A Review of Indirect Tool Condition Monitoring Systems and Decision-Making Methods in Turning: Critical Analysis and Trends’, *Sensors*, vol. 21, no. 1, p. 108, 2020. https://doi.org/10.3390/s21010108

[3] F. Halila, C. Czarnota, and M. Nouari, ‘Analytical stochastic modeling and experimental investigation on abrasive wear when turning difficult to cut materials’, *Wear*, vol. 302, no. 1–2, pp. 1145–1157, 2013. https://doi.org/10.1016/j.wear.2012.12.055

[4] P. Muthuswamy and S. K, ‘Artificial intelligence based tool condition monitoring for digital twins and industry 4.0 applications’, *Int. J. Interact. Des. Manuf. IJIDeM*, vol. 17, no. 3, pp. 1067–1087, Jun. 2023. https://doi.org/10.1007/s12008-022-01050-5

[5] L. Colantonio, L. Equeter, P. Dehombreux, and F. Ducobu, ‘A Systematic Literature Review of Cutting Tool Wear Monitoring in Turning by Using Artificial Intelligence Techniques’, *Machines*, vol. 9, no. 12, p. 351, Dec. 2021. https://doi.org/10.3390/machines9120351

[6] R. U. Baig, S. Javed, M. Khaisar, M. Shakoor, and P. Raja, ‘Development of an ANN model for prediction of tool wear in turning EN9 and EN24 steel alloy’, *Adv. Mech. Eng.*, vol. 13, no. 6, p. 168781402110267, Jun. 2021. https://doi.org/10.1177/16878140211026720

[7] D. Y. Pimenov, A. Bustillo, S. Wojciechowski, V. S. Sharma, M. K. Gupta, and M. Kuntoğlu, ‘Artificial intelligence systems for tool condition monitoring in machining: analysis and critical review’, *J. Intell. Manuf.*, vol. 34, no. 5, pp. 2079–2121, Jun. 2023. https://doi.org/10.1007/s10845-022-01923-2

[8] L. Colantonio, L. Equeter, P. Dehombreux, and F. Ducobu, ‘Comparison of cutting tool wear classification performance with artificial intelligence techniques’, pp. 1265–1274, May 25, 2023.

[9] A. J. Bishara and J. B. Hittner, ‘Testing the significance of a correlation with nonnormal data: Comparison of Pearson, Spearman, transformation, and resampling approaches.’, *Psychol. Methods*, vol. 17, no. 3, pp. 399–417, Sep. 2012. https://doi.org/10.1037/a0028087

[10] T. Rabczuk and K.-J. Bathe, Eds., *Machine Learning in Modeling and Simulation: Methods and Applications*. in Computational Methods in Engineering & the Sciences. Cham: Springer International Publishing, 2023. https://doi.org/10.1007/978-3-031-36644-4

[11] H. Li, *Machine Learning Methods*. Singapore: Springer Nature Singapore, 2024. https://doi.org/10.1007/978-981-99-3917-6