Available online at www.eccomasproceedia.org
Eccomas Proceedia SMART (2023) 1041-1052

ECCOMAS
Proceedia

X ECCOMAS Thematic Conference on Smart Structures and Materials
SMART 2023
D.A. Saravanos, A. Benjeddou, N. Chrysochoidis and T. Theodosiou (Eds)

TOOLS WEAR MONITORING DURING THE TURNING PROCESS WITH COMPUTATIONAL INTELLIGENCE APPROACHES

A. NEMATI[1], R. PARDOE[3], M. PANESSO[2], S. F. BOCKLISCH[1] AND W.G.DROSSEL[1,2]

[1] Chemnitz University of Technology
Reichenhainer Str. 70, 09126, Chemnitz, Saxony, Germany
e-mails: amir.nemati@mb.tu-chemnitz.de
steffen.bocklisch@etit.tu-chemnitz.de
Welf-Guntram.Drossel@iwu.fraunhofer.de

[2] Fraunhofer Institute for Machine Tools and Forming Technology
Noethnitzer Straße 44, 01187 Dresden, Germany
emails: Miguel.Antonio.Panesso.Perez@iwu.fraunhofer.de
Welf-Guntram.Drossel@iwu.fraunhofer.de

[3] Northwestern university
633 Clark St, Evanston, IL 60208, USA
email: robertpardoe2024@u.northwestern.edu

Abstract. Tool wear monitoring and prediction are significant in advanced products for developing intelligent manufacturing systems. In smart manufacturing, the aim is to improve the production quality of output and reduce costs. Tool wear significantly impacts the quality of the surface finish and the dimensional accuracy of the workpiece. Correspondingly, using worn tools may result in poor surface quality or even in tool breakage, leading to unplanned downtime. Tool wear monitoring techniques are divided into direct and indirect categories. The direct method focuses on the wear measurement technique after the process, and there is a significant time delay in the machining process. Alternatively, in the indirect method, the amount of tool wear associated with process parameters and time-varying data, such as force, is measured. This method is compatible within the production line, but faces the challenge of is the complicated and expensive calculations to achieve an estimated model. In such situations, a data-driven black-box nonlinear system provides a reasonable approximation and an efficient approach. The approach examined in this article consists of a hybrid method (direct and indirect) to measure data (here force) and use computational intelligence such as fuzzy pattern classification and neural networks for modeling. The studied tools are single-edged, and the measured data is obtained during the turning process.

Key words: Tool condition monitoring, Fuzzy membership functions, Recurrent neural network, Long short-term memory.

1 INTRODUCTION

Intelligent monitoring is essential to the operation, maintenance, and optimal scheduling of production processes in the context of Industry 4.0 and intelligent manufacturing systems. In

Peer-review under responsibility of the organizing committee of SMART 2023.
doi: 10.7712/150123.9855.444298

smart manufacturing, a factory can monitor a production process through data monitoring and automation services to improve the production quality of output and reduce costs. Tool wear significantly impacts the quality of the surface finish and the product's shape accuracy. Likewise, using worn tools may result in tool breakage, leading to unplanned downtime. Therefore, waste products can be avoided by substituting worn tools in time. In addition, by correctly and accurately predicting the tool's life, the cost of tools can be significantly reduced. Furthermore, when the tool wear is estimated in time, it is possible to adjust the tool position to guarantee the quality of a particular workpiece surface [1]–[3]. Tool wear monitoring techniques are divided into direct and indirect categories. The direct method focuses on the wear measurement technique after the process. In this method, the cutting process must be interrupted, and the machining process has a significant time delay. However, in the indirect method, the amount of tool wear associated with process parameters and time-varying data, such as force, is measured. This method is easier to measure, but the main challenge is the complicated and expensive calculations to achieve an estimated model [4]. In such situations, a data-driven black-box nonlinear system provides a reasonable approximation and an efficient approach [5]. The approach examined in this article is the hybrid method (direct and indirect) to measure data (here force) and use computational intelligence for modeling.

2 PROBLEM STATEMENT

Reducing tool wear in industrial production systems is one of the most critical issues in machining operations. The nature of the wear phenomenon is due to the effect of many factors on the cutting tool, which mainly reflect on the cutting force and on the temperature development. These are dependent on cutting conditions like cutting speed, cutting depth feed rate, cooling (or the absence of it), material of the workpiece and material of the tool. These factors lead to the wear phenomenon because plastic shearing of the workpiece material and friction in the contact zone increases the temperature of the cutting tools and causes the tool to wear. Failure to carefully check the wear of the tools causes high costs during production of workpieces. Tool wear can cause parts to fall outside of manufacturing tolerances requiring them to be reworked. It is also independently costly to replace the tools. In fact, in this phenomenon, the material can no longer be reused, and therefore, the costs related to wear in the industry are high. In the last decades, many efforts have been made to increase reliability and reduce the cost of tool condition monitoring systems (TCMs). There are online and offline approaches to TCM. In the offline approach, work must be stopped to take a discrete measurement of the state of the tool or workpiece. In the online or sensor-based approach, tool wear monitoring can be divided into two main groups, direct and indirect. The direct methods measure the amount of tool wear in a specific area, at the point of contact between the tool and the workpiece, with a camera. These approaches are almost impossible due to the environmental conditions around the machine tool, such as chips and coolant; they are costly and prone to errors. In the indirect approach, process parameter signals related to tool wear (such as force) are measured and extracted. This approach aims to develop functional based upon parameters of the turning processes in order accurately predict the state of tool wear. Modelling tool wear is a complex procedure with many influencing factors are which are either not measurable or unavailable, such as the friction force between the tool and the workpiece. Therefore, tool wear is a phenomenon that has nonlinear behavior and complexity. In this situation, the lack of information and detailed knowledge about the system makes it impossible to have a physical or accurate theoretical model. Additionally, existing theoretical models are imprecise because the input data are simplified to linearly model a nonlinear system. Hence, a certain amount of inaccuracy and uncertainty can be seen in these models. In such a situation where a system is nonlinear, the data-driven black box provides a reasonable approximation of dynamic systems. These approaches build models based on measured process input and output data and require little or no physical information. Therefore, black box models, such as models based on fuzzy logic and neural networks instead of statistical models and mathematical equations, are widely used to model the behavior of nonlinear systems. The Fuzzy Pattern Classification (FPC) method is a fundamental model-building method recommended widely. FPC is suitable for describing complex relationships based on data-based or expert knowledge. The formal description of classes is done by membership functions (MF), which can be performed in one or more dimensional information spaces (attribute, quality, or state space). The present work aims to model based on the measured data, in other words, force (see section 4) to predict the tool's wear. Therefore, the data set is modeled with different computational intelligence methods, such as FPC, multilayer perceptron (MLP), long short-term memory (LSTM), and bidirectional LSTM (BiLSTM). The results of each model are evaluated.

3 MODELING METHODOLOGIES

3.1 Fuzzy Classification

The fuzzy classification is an effective option for evaluating complex systems when the theoretical models are insufficient or impractical. The fuzzy classification method was derived from the idea of fuzzy sets [6]. Unlike in typical sets where set membership is binary, in fuzzy sets, objects have degrees of membership to a given set represented by a numerical value between 0 and 1. A given object's membership to a set is evaluated/determined using a membership function (MF) corresponding to that set. In the fuzzy classification approach, the given signals (time series or sequence data) or samples (discrete data) are members of the various fuzzy sets, acting as objects. Fuzzy classification can evaluate observations represented as feature vectors in a Euclidian feature space of any number of dimensions. Fuzzy sets in this space represent the possible classification/classes. Fuzzy classification can be conducted in a data-driven or a knowledge-based approach on a class-by-class basis through the adaptation of MFs [7] .What is chosen as the MF depends on the situation but is often somewhat subjective. Triangular, trapezoidal, and Gaussian functions are often chosen for Fuzzy MFs in the literature. A modified Aizerman potential function [7], [5], [8], [9] is another choice, and it has a number of properties that can make it preferential to other functions. Unlike the trapezoidal and triangular functions, the potential function is nonlinear, and unlike the Gaussian function, it can be asymmetrical. The potential function is highly customizable through eight parameters. The following sections discuss one and multi-dimensional uses of the potential function as an MF.

3.1.1 One-Dimensional Membership Function

First, we consider an MF in a fuzzy classification problem in a one-dimensional feature space. We use the asymmetric version of potential function, shown below in equation (1), as a function of the variable u as our MF. Here the membership function for each class is evaluated, and the higher membership value is determined to be the class of that window. Therefore, the output of the MFs for each step has been used for the decision of the final output (see section 5).

$$u) = \begin{cases} \dfrac{a}{1+\left(\dfrac{1}{b_l}-1\right)\left(\dfrac{r-u}{c_l}\right)^{d_l}} & u < r \\ \dfrac{a}{1+\left(\dfrac{1}{b_r}-1\right)\left(\dfrac{u-r}{c_r}\right)^{d_r}} & u \geq r \end{cases} \tag{1}$$

The asymmetric function has eight parameters where parameters b, c, and d are divided into left and right values. The parameter r determines the position of the MF and the location of its peak value. The parameter a ($a >= 1$) prescribes a maximum value to the membership function and scales the output of the membership function; the membership function outputs a when its input value is $u = r$. The parameters c_l and c_r $\left(c_{l/r} > 0\right)$ assign the distance of the function's inflection points below and above the mean, respectively. They are defined by the difference between the minimum and maximum values of the set from the mean. The parameters b_l and b_r $\left(0 < b_{l/r} \leq 1\right)$ are multipliers that determine the value of the MF at its inflection as a fraction of the parameter a. The parameters d_l and d_r $\left(d_{l/r} \geq 2\right)$ determine the shape of the function and the distribution of objects within the class represented by the MF. The effects of the parameters are shown in figure (1).

3.1.2 Multi-Dimensional Membership Function

The potential function is able to be defined for any number of features N. A multi-dimensional MF can be formed from several one-dimensional MFs so that it is defined in the N-dimensional feature space $\mathbb{R}^N$. A unique advantage of the potential function is that a combination of multiple one-dimensional MFs is possible. This is accomplished using the N-fold compensatory Hamacher average operator. For a multi-dimensional MF, there is a set of parameters r, $b_{l/r}$, $c_{l/r}$, and $d_{l/r}$ unique to each of the N dimensions in the feature space; however, a single parameter defines the peak value at the $a = 1$ of the function. This multi-dimensional membership function's form is illustrated in equations (2) and (3).

$$(\underline{u}) = \frac{1}{1+\left(\frac{1}{N}\cdot\sum_{n=1}^{N}\sigma_n\right)} \tag{2}$$

where n belonging to number of features ($n = 1, 2, 3, \ldots, N$; N = the number of dimension), Therefore σ_n denoted by

$$_n = \begin{cases} \left(\dfrac{1}{b_{l,n}} - 1\right)\left(\dfrac{r_n - u_n}{c_{l,n}}\right)^{d_{l,n}} & u_n < r_n \\ \left(\dfrac{1}{b_{r,n}} - 1\right)\left(\dfrac{u_n - r_n}{c_{r,n}}\right)^{d_{r,n}}, & u_n \geq r_n \end{cases} \tag{3}$$

A two-dimensional feature space containing four membership functions with varying parameters is shown in figure (2).

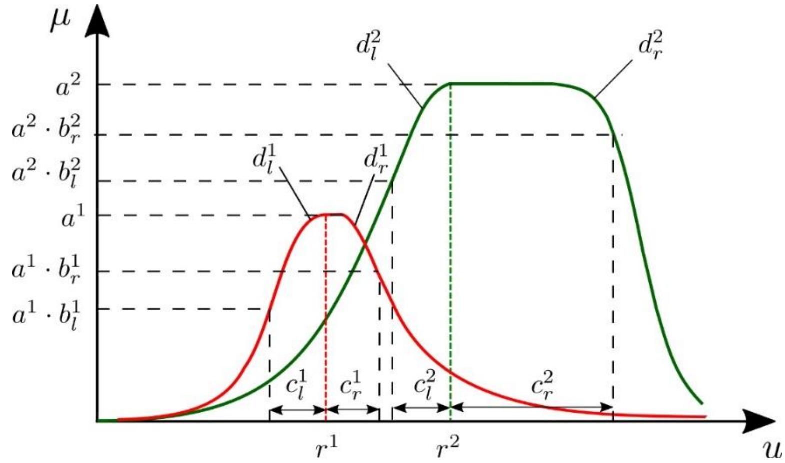

Fig. 1 Two one-dimensional asymmetric membership functions with different parameters [5]

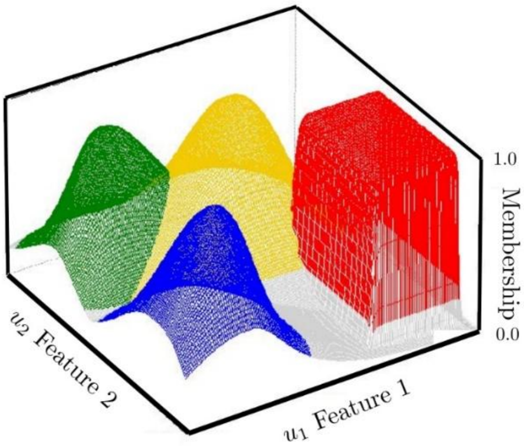

Fig. 2 Illustration of two-dimensional MFs for four classes [5]

3.2 Multilayer perceptron (MLP)

Artificial neural networks (ANNs) such as MLPs are modeling structures consisting of many interconnected artificial neurons, also called nodes. ANNs are fundamentally like any other black box model in that they take an input vector, conduct various operations, and produce an output vector. Any ANN consists of an input layer, at least one hidden layer, and an output layer. Each neuron takes inputs and produces a single output or activity; this output can be passed as an input to neurons in the next layer over the edges. Edges have weights associated with them, the output of a neuron is a function of its weighted inputs, and the function used is called the activation function. The activation function is nonlinearly allowing ANNs to approximate complex relationships [10].

3.3 Long short-term memory (LSTM)

Recurrent neural networks (RNNs) are a type of ANN designed to model datasets in which there are interdependencies between data points, such as time series. RNNs are chain-like in their construction; each neuron takes as input the current time step data as well as the previous neuron's state and produces an output. There is a frequently encountered problem with the training of these networks called the vanishing gradient problem. LSTM is a type of recurrent neural network proposed by Hochreiter in 1997 [11] to avoid the vanishing gradient problem. LSTM networks keep the same broad chain-like structure as traditional RNNs; however, the single neuron for each step is replaced by a more complex module [12], [13].

3.4 Bidirectional Long Short-Term Memory (BiLSTM)

BiLSTM is an extension of the LSTM structure; two independent chains of modules, or layers, intake input data in opposite sequence order. In essence, in one chain state, information from the previous module is passed forward, while in the other state, information from the next module is passed back. Both layers then feed into a single output layer [14].

4 DATA COLLECTION AND CUTTING EXPERIMENTS

Cutting experiments were performed, during which the cutting force was continuously measured with a piezoelectric sensor located on the cutting tool. The main criteria for the sensor integration consisted in minimizing the required effort for adapting the tool-holder to the geometry of commercially available piezoelectric sensors. To allow an integration with minimal sensor size, a $12\,mm$ quartz-sensor of the 9132B model series of the company Kistler was used in two positions of a DDJNL 2020K 15 tool-holder of the manufacturer Sandvik (see figure(3))

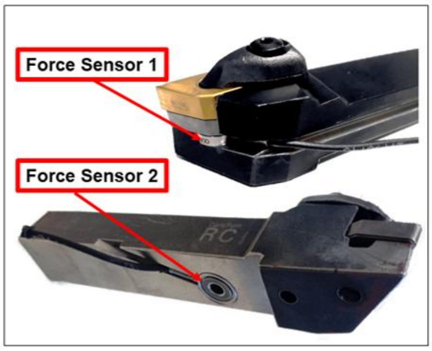

Fig. 3 Integration of SlimeLine sensors (Kistler) in two locations of the tool-holder (Sandvik)

4.1 TESTING METHODS

The concepts for integrating the SlimLine sensors as were tested in two scenarios in which the tool inserts were worn until failure. In both cases, a cylinder bar of steel alloy $\mathbf{Cr42Mo4}$ of $\mathbf{65}\,\boldsymbol{mm}$ in diameter and $\mathbf{200}\,\boldsymbol{mm}$ length was reduced during the tests to a diameter $\mathbf{20}\,\boldsymbol{mm}$ under dry conditions at cutting depths of $\mathbf{1}\boldsymbol{mm}$ and $\mathbf{1.5}\,\boldsymbol{mm}$, a feed rate of $\mathbf{0.3}\,\boldsymbol{mm/rev}$ and a cutting speed of $\mathbf{270}\,\boldsymbol{m/min}$. In the first scenario, a single force signal was acquired for the total length of the cylinder bar with a new measurement for each successive smaller diameter. A graphical explanation of the method is given in figure (4) and figure (5).

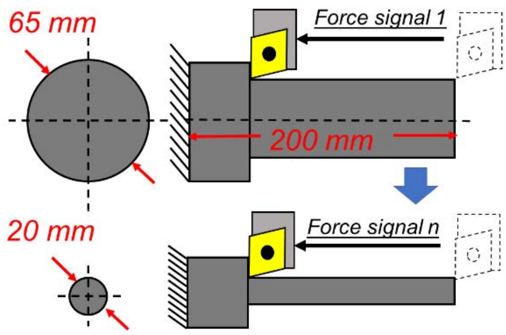

Fig. 4 Measuring method 1 for tool life tests consisting of longer cutting lengths

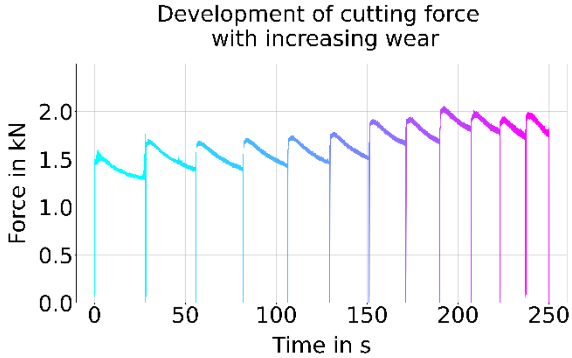

Fig. 5 Measuring Force signals corresponding to span of tool life following experimental method of Fig. 4

In terms of the signal acquisition system and the measuring parameters, all force sensors were connected to a charge amplifier type 5011A from Kistler which in turn was connected to a M410B QuantumX measuring platform. The sampling rate was set to 4.8 kHz. During the cutting experiments, the state of tool wear was documented at several stages of the tool life by means of a USB-microscope AM4515ZT Dino-Lite Edge with a magnification ratio of 100x and a resolution of 1280 x1024 pixels. An example documenting three stages is presented in figure (6).

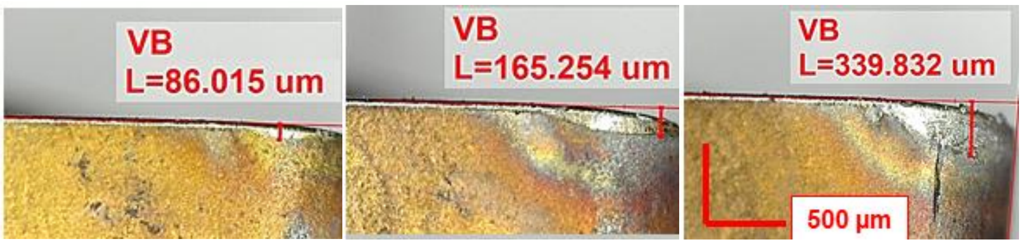

Fig. 6 Visualizing tool wear: from minimal (86 µm) to significant (340 µm) on the tool insert during cutting experiments

5 RESULTS

In this section, the specific execution of the methods discussed in section 3, the data processing, and the experimental results are presented. All models were implemented, trained, and tested in the MATLAB programming environment.

5.1 Data Processing

The data collected consisted of 98 time period segment with an average length of 94 thousand time steps. The time period segment were measures of the force on the cutting bit or turning insert. Additionally, each signal has an associated wear value. Based on this wear value, the signals were split into two classes, wear $<300\mu m$ as class 1 and wear$>300\ \mu m$ as class 2. The distinction at $300\mu m$ is based on expert evaluation. This value could vary depending on the situation, what is being manufactured, and the tool being used. The 98 time period segment came from five different turning operations, with each turning operation being carried out at one of three different depths of cut ($1mm$, $1.5mm$, and $2mm$). This information was not taken into account when modeling as it would make the available data for each class too limited. After assigning classes, the data were then stratified and split into training and testing sets with the K-fold method using 3-folds [5]. For the static approach, the signals were then each truncated to the same size, meaning that for each signal, all data points after that, one corresponding to the last data point of the shortest signal, were removed. For the time series approach, since the model is trained using batch learning, the data was not truncated. Then for the static approach as a hyperparameter, the signals were all split into N windows. This is to say, subdivided into N sections containing an equal number of time steps with the exception of the last window. Whereas for the LSTM approaches, since the signals were not truncated to an equal length, the number of windows into which each signal was divided varied depending on signal length. A single window length T was defined as a hyperparameter in hertz or windows per second. Every signal was divided into multiple segments of the specified length, with the last segment being marginally longer when necessary to capture the whole signal. For both approaches, the mean force value across each window and the standard deviation of the force values within each window were then calculated for each of the windows for each of the 98 time period segment. This was the final data set used for training consisting of 98 time period segment with two featureswith varying signal lengths for the LSTM approach and constant signal lengths for the static approach.

5.2 Comparison accuracy of various modeling

After data processing, we have several observations for any given signal with two features: mean value as the first feature and standard deviation as the second feature figure (7 left). Fuzzy pattern classification meant that a different set of parameters defined a membership function for each class specific to each window for each K-fold. Regarding the hyperparameters, for all MFs $\boldsymbol{b_{l/r}}$ was set to $\mathbf{0.5}$, parameter $\boldsymbol{a}$ was set to $\mathbf{1}$, parameters $\boldsymbol{c_{l/r}}$ and $\boldsymbol{r}$ were determined from the signal data, and parameter $\boldsymbol{d}$ was a hyperparameter. With the MF parameters determined, the classification procedure of a test signal was/follows: the signal is split into windows, and the features are determined as described above. Then going across each signal window, the membership function for each class is evaluated at the value of the features. Whichever function produces the higher membership value is determined to be the class of that window; a class value of one or two is stored accordingly. Additionally, the output of the MFs is stored for possible later reference. Once these calculations have been made for all windows, which-ever class the windows were determined to be more frequently is then predicted as the class of the signal. If there are 100 windows and the class value is determined to be one 51 times or more, then the signal is predicted to be a member of class 1. If, for example, both class values occurred 50 times for a 100 window signal, the procedure is as follows: sum the class 1 membership values over all windows where class 1 was predicted and independently sum all the class 2 membership values over the windows where class 2 was predicted. In the example of 100 windows, you would sum the membership values over 50 windows for each class. Whichever class has the greater sum is predicted to be the class of the signal (see figure (8)).

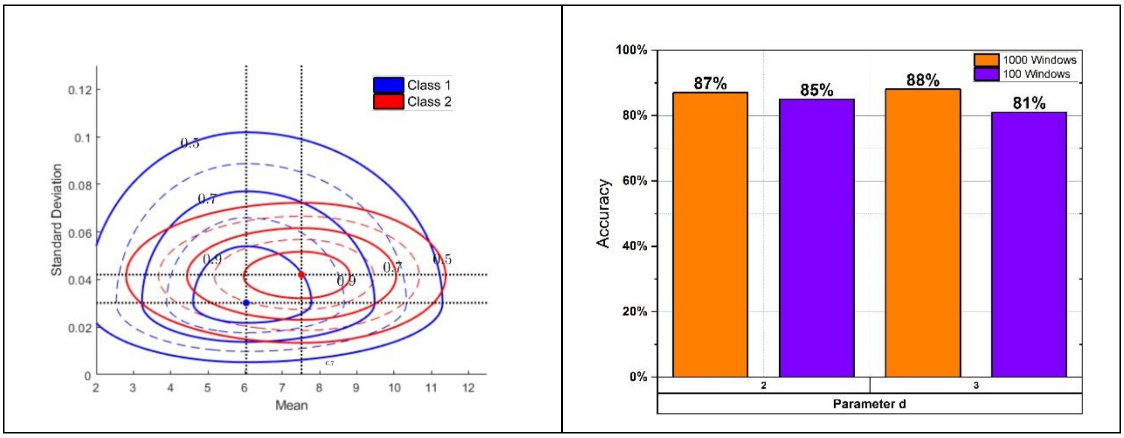

Fig. 7 Fuzzy pattern classification with two features for two classes with different α –cuts of 0.9, 0.7 and 0.5 (left) and accuracy value of the fuzzy model with two settings for parameter d (right)

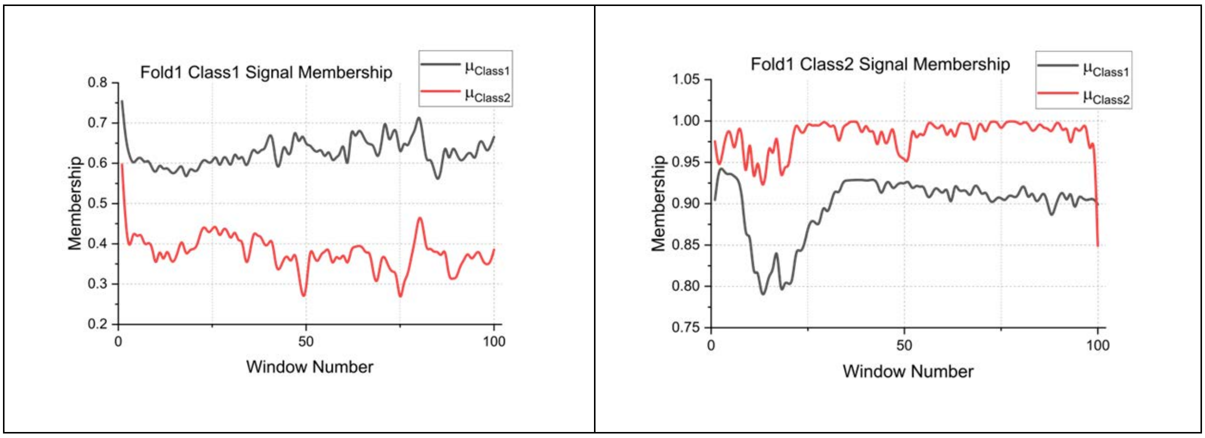

Fig. 8 Comparison between two classes of normal (left) and wear signals (right)

As shown in figure (7-right), the accuracy value of the fuzzy model with d parameters for the values set in $\boldsymbol{d = 2}$ and $\boldsymbol{d = 3}$ in the window value of 1000 is almost equal, and this difference is about 1.14%. While the value of accuracy in window 100 and for parameter $\boldsymbol{d = 2}$ is equal to $\mathbf{0.85}$, and the value of 4.71% shows the difference with the value of parameter $\boldsymbol{d = 3}$. Therefore, it can be evaluated from the fuzzy model that the parameter $\boldsymbol{d = 3}$ with a window of 1000 shows more accuracy. The comparison of the fuzzy, MLP, and LSTM models with Bi-LSTM as a primary reference is shown in Figure (9). In the MLP model, the number of hidden layers is 17, and the window value is 1000; as seen in Figure (9), the value accuracy of the MLP model is $\mathbf{4.35\%}$ more accurate than Bi-LSTM. The value of the accuracy of the fuzzy model expresses more accuracy for this problem, and this value is $\mathbf{27.54}$ % more accuracy than Bi-LSTM. Figure (9) also shows the accuracy of the

LSTM and Bi-LSTM models. In evaluating the accuracy of these models, LSTM performs better with 144 hidden layers, and this model shows $\mathbf{5.8}\%$ more accuracy than Bi-LSTM.

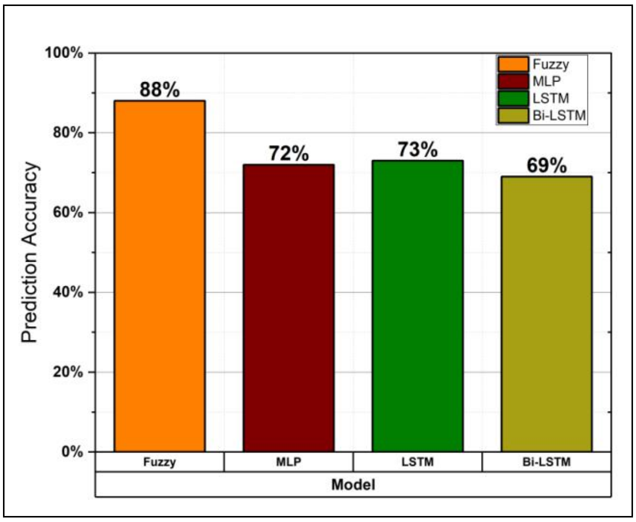

Fig. 9 Comparison accuracy between various modeling approaches from force signal to predict tool wear

6 CONCLUSIONS

The tool wears monitoring and prediction are crucial in developing intelligent manufacturing systems for advanced products. The impact of tool wear on surface finish quality and dimensional accuracy of workpieces must be considered. The use of worn tools leads to unplanned downtime, affecting productivity. However, tool wears monitoring techniques face challenges, as they are divided into direct and indirect categories with complex and costly calculations required for accurate estimation models. In such scenarios, a data-driven black-box nonlinear system offers a practical approximation and an efficient approach. The hybrid method discussed in this article combines direct and indirect measurements by utilizing computational intelligence techniques such as fuzzy pattern classification and neural networks for modeling. The study focuses on single-edged tools, and the data collected during the turning process consists of $\mathbf{98}$ time period segments, each with an average length of $\mathbf{94}$ thousand time steps. Based on the wear value, the collected signals were categorized into two classes: wear $<\mathbf{300}\boldsymbol{\mu m}$ µm as class $\mathbf{1}$ and wear > 300 µm as class $\mathbf{2}$. Among the different models evaluated, the fuzzy model achieved the highest accuracy. Notably, the best results were observed for $\boldsymbol{d} = \mathbf{2}$ and $\boldsymbol{d} = \mathbf{3}$ values when the window value was set to $\mathbf{1000}$. This indicates the effectiveness of the proposed approach in accurately estimating tool wear and facilitating timely maintenance actions to minimize unplanned downtime and optimize manufacturing processes.

REFERENCES

[1] G. Totis and M. Sortino, “Development of a modular dynamometer for triaxial cutting

force measurement in turning," International Journal of Machine Tools and Manufacture, vol. 51, no. 1, pp. 34–42, 2011, doi: https://doi.org/10.1016/j.ijmachtools.2010.10.001.

[2] G. Totis, G. Wirtz, M. Sortino, D. Veselovac, E. Kuljanic, and F. Klocke, "Development of a dynamometer for measuring individual cutting edge forces in face milling," Mechanical Systems and Signal Processing, vol. 24, no. 6, pp. 1844–1857, 2010, doi: https://doi.org/10.1016/j.ymssp.2010.02.010.

[3] C. Scheffer, H. Kratz, P. S. Heyns, and F. Klocke, "Development of a tool wear-monitoring system for hard turning," International Journal of Machine Tools and Manufacture, vol. 43, no. 10, pp. 973–985, 2003, doi: https://doi.org/10.1016/S0890-6955(03)00110-X.

[4] L. Dan and J. Mathew, "Tool wear and failure monitoring techniques for turning—A review," International Journal of Machine Tools and Manufacture, vol. 30, no. 4, pp. 579–598, 1990, doi: https://doi.org/10.1016/0890-6955(90)90009-8.

[5] A. Nemati, M. Jäckel, S. F. Bocklisch, and W.-G. Drossel, "Fuzzy pattern modeling of self-pierce riveting for data from experiments and computer simulations," Int J Adv Manuf Technol, vol. 122, no. 5–6, pp. 2203–2216, Sep. 2022, doi: 10.1007/s00170-022-09914-6.

[6] L. A. Zadeh, "Fuzzy sets," Information and Control, vol. 8, no. 3, pp. 338–353, 1965, doi: https://doi.org/10.1016/S0019-9958(65)90241-X.

[7] S. F. Bocklisch, Prozessanalyse mit unscharfen Verfahren. Verlag Technik Berlin, 1987.

[8] M. Baur, "Linear Local Models as Local Approximators," p. 6.

[9] F. Bocklisch, S. F. Bocklisch, and J. F. Krems, "Sometimes, often, and always: Exploring the vague meanings of frequency expressions," Behav Res, vol. 44, no. 1, pp. 144–157, Mar. 2012, doi: 10.3758/s13428-011-0130-8.

[10] M. Paliwal and U. A. Kumar, "Neural networks and statistical techniques: A review of applications," Expert Systems with Applications, vol. 36, no. 1, pp. 2–17, 2009, doi: https://doi.org/10.1016/j.eswa.2007.10.005.

[11] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[12] Y. Yu, X. Si, C. Hu, and J. Zhang, "A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures," Neural Computation, vol. 31, no. 7, pp. 1235–1270, Jul. 2019, doi: 10.1162/neco_a_01199.

[13] K. Smagulova and A. P. James, "A survey on LSTM memristive neural network architectures and applications," Eur. Phys. J. Spec. Top., vol. 228, no. 10, pp. 2313–2324, Oct. 2019, doi: 10.1140/epjst/e2019-900046-x.

[14] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF Models for Sequence Tagging." arXiv, Aug. 09, 2015. Accessed: Mar. 20, 2023. [Online]. Available: http://arxiv.org/abs/1508.01991