The International Journal of Advanced Manufacturing Technology
https://doi.org/10.1007/s00170-023-10811-9

ORIGINAL ARTICLE

Tool remaining useful life prediction using bidirectional recurrent neural networks (BRNN)

Telmo Fernández De Barrena[1] · Juan Luís Ferrando[1] · Ander García[1] · Xabier Badiola[2] · Mikel Sáez de Buruaga[2] · Javier Vicente[2]

Received: 8 March 2022 / Accepted: 12 October 2022

Abstract

Nowadays, new challenges around increasing production quality and productivity, and decreasing energy consumption, are growing in the manufacturing industry. In order to tackle these challenges, it is of vital importance to monitor the health of critical components. In the machine tool sector, one of the main aspects is to monitor the wear of the cutting tools, as it affects directly to the fulfillment of tolerances, production of scrap, energy consumption, etc. Besides, the prediction of the remaining useful life (RUL) of the cutting tools, which is related to their wear level, is gaining more importance in the field of predictive maintenance, being that prediction is a crucial point for an improvement of the quality of the cutting process. Unlike monitoring the current health of the cutting tools in real time, as tool wear diagnosis does, RUL prediction allows to know when the tool will end its useful life. This is a key factor since it allows optimizing the planning of maintenance strategies. Moreover, a substantial number of signals can be captured from machine tools, but not all of them perform as optimum predictors for tool RUL. Thus, this paper focuses on RUL and has two main objectives. First, to evaluate the optimum signals for RUL prediction, a substantial number of them were captured in a turning process and investigated by using recursive feature elimination (RFE). Second, the use of bidirectional recurrent neural networks (BRNN) as regressive models to predict the RUL of cutting tools in machining operations using the investigated optimum signals is investigated. The results are compared to traditional machine learning (ML) models and convolutional neural networks (CNN). The results show that among all the signals captured, the root mean squared (RMS) parameter of the forward force (F_y) is the optimum for RUL prediction. As well, the bidirectional long-short term memory (BiLSTM) and bidirectional gated recurrent units (BiGRU), which are two types of BRNN, along with the RMS of F_y signal, achieved the lowest root mean squared error (RMSE) for tool RUL, being also computationally the most demanding ones.

Keywords Tool remaining useful life prediction · Machine tool · Prognostics · Long-short term memory network · Bidirectional recurrent neural network · Deep learning

✉ Telmo Fernández De Barrena
tfernandez@vicomtech.org

Juan Luís Ferrando
jlferrando@vicomtech.org

Ander García
agarcia@vicomtech.org

Xabier Badiola
xbadiola@mondragon.edu

Mikel Sáez de Buruaga
msaez@mondragon.edu

Javier Vicente
jvicentet@mondragon.edu

[1] Fundación Vicomtech, Basque Research and Technology Alliance (BRTA), Mikeletegi 57, 20009 Donostia-San Sebastian, Spain

[2] Faculty of Engineering, Mondragon University, 20500 Mondragon, Spain

Published online: 01 February 2023

1 Introduction

Nowadays, the manufacturing industry is trying to achieve an optimum scenario of production. That scenario includes working to improve the productivity, enhance the reliability and availability of engineering services, and reduce the costs and downtime on production [1]. According to [2], 20% of the time of machine tool downtime is due to tool failure, leading to a reduction of productivity and economic losses. For that reason, to fulfill that goal, this industry is working on different predictive maintenance strategies to optimize the availability of the machine tools in the manufacturing industry. These strategies include monitoring the health of the components to build data-based models for tool wear diagnosis and prognosis. Tool wear diagnosis attempts to monitor the actual tool wear, whereas the prognosis aims to predict the future wear of the tool.

Tool wear is generated as a result of chemical, thermal, and mechanical interactions between the tool and workpiece materials. These interactions are the cause of the two main types of tool wear that can define the end of tool-life: flank wear and crater wear [3]. The effectiveness of the process is commonly linked to the degree of flank wear. For that reason, this variable is usually taken as a wear indicator for the industry [4]. As the tool wear affects several critical characteristics, such as the surface roughness and dimensional accuracy of the manufactured elements, it is a particularly important feature to monitor in machining processes. Besides, it is known that the amount of energy needed to manufacture a piece directly depends on the degree of tool wear [5].

Remaining useful life (RUL) is usually defined as the time remaining for a component to stop performing its functional capabilities. In contrast to the tool wear diagnosis, the estimation of RUL (or time-to-failure) of a component or system is a prognostic method. Prognostics are gaining importance over diagnosis in the machinery industry, as they help to develop optimum maintenance strategies and optimize time and resources.

There are several prognostics prediction methods used for determining the RUL of subsystems or components, which can be divided in three groups: physics-based, data-based, and hybrid [6]:

Physics-based These methods consist in building mathematical models based on the failure mechanisms or the first principle of damage [7]. They are relatively accurate and precise models and easy to validate. However, they require an extensive prior knowledge about physical systems, which is usually unavailable in practice [8]. Thus, building this kind of models for RUL prediction is expensive and time consuming, computationally intensive, and with a risk of not achieving the desired outcome [9].

Data-based In these methods, statistics and computational intelligence approaches are used for RUL prediction. Although the accuracy and precision of physics-based methods are generally higher than data-based ones, they are also more difficult to obtain. In contrast, data-driven models usually require sufficient historical data for training models and do not rely on much prior expertise on prognostics. For that reason, many data-driven algorithms have been proposed in the recent years and good prognostic results have been achieved [8].

Hybrid These types of methods try to combine the advantages of both physic-based and model-based approaches. However, due to the complexity and time needed to develop both models, most of the researchers usually just develop one of them, and a relatively small number of researchers have focused their attention on the hybrid approaches [10].

Among the data-based models, machine learning (ML) models can be outstood. ML is an evolving branch of computational algorithms designed to emulate human intelligence by learning from the surrounding environment. To feed the data-based ML models, different signals must be captured. In the industrial machining environment, usually force, acoustic emission (AE), and electric current signals are used as source of data [11]. Diei and Dornfeld [12] studied both the cutting forces and AE signals are closely correlated with the tool wear condition of a milling machine. Ferrando et al. [4] also showed that AE signals with their corresponding preprocessing can predict tool flank wear. On the other hand, cutting speed, feed, and the depth of cut are the cutting parameters with most influence on the wear of the tools [13].

Along with the improvement of computational infrastructure, deep learning (DL) has become one of the main research topics in the field of prognostics. DL is one of the sub-branches of ML, having the capacity to capture the hierarchical relationship embedded in deep structures [14]. In recent years, the literature around this topic has considerably grown, showing an increasing interest and suggesting a promising future of DL in RUL prediction. DL RUL prediction approaches are purely data-driven approaches. For this reason, to obtain trustworthy models, a large database of run-to-fail trajectories should be obtained and compared to the observed data [6].

RUL prediction is a well-known topic which have a considerable number of papers. Schwabacher [15] was the first author to work around this issue back in 2005. In the field of tool wear diagnosis and prognostics, some of the popular ML architectures are the artificial neural networks (ANNs), which includes single/multilayer perceptron (MLP), convolutional neural network (CNN), and recurrent neural network (RNN):

Single/multilayer perceptron Single-layer perceptron network is the simplest kind of neural network. It is a single-layer feed-forward ANN. The inputs are connected directly with the output, each connection containing an associated weight. The multilayer perceptron (MLP) follows the same idea, containing multiple layers instead of just one. Each neuron in one layer has directed connections to the neurons of the subsequent layer.

CNN CNN is a multi-layer feed-forward ANN, first put forward by Lecun et al. [16]. It focuses on two-dimensional inputs, using stacking convolutional layers and pooling layers to achieve the features learning [6]. It is well known for its capability to reveal abstract visual features. Whereas the CNN-based approaches have achieved excellent results in the machinery fault diagnosis and surface integration inspection [17], there are few research reports on their application in RUL prediction.

RNN RNN is a deep architecture that contains feedback connections from hidden or output layers to the preceding layers, being able to process dynamic information [18]. It can model time sequence data [19]. Some work [20] applied RNN for RUL estimation. However, RNN is known to have long-term time dependency problems [21], where the gradients propagated over many stages tend to either vanish or explode. For this reason, long-short term memory (LSTM) network has been developed. It controls the information flow using an input gate, forget gate, and output gate, being able to solve the long-term time dependency issues [22]. LSTM [23] is a type of RNN network for sequence learning tasks and has achieved remarkable results on speech recognition and machine translation. Gated recurrent units (GRUs) are like LSTM units, but instead of having 3 gates, they contain 2, the reset and update gates. They use less training parameters and therefore less memory.

ANNs have been used in different scenarios, such as in bearings RUL prediction, RUL prediction in simulated and benchmark datasets, and tool wear prediction. Huang et al. [24] utilized the traditional MLP approach for predicting the RUL of the laboratory-tested bearings and concluded that the results were better than reliability-based approaches. Tian [25] developed an ANN method for estimating the RUL of equipment, taking the age and multiple condition monitoring measurement values at the present and previous inspection points as the inputs, and the equipment life percentage as the output. Li et al. [8] developed a DL method for prognostics based on CNNs. Dropout technique was employed to avoid overfitting problem. The popular C-MAPSS synthetic dataset was used. With raw feature selection, data pre-processing, and sample preparation using time window, good prognostic performance was achieved with the proposed method, outperforming narrowly LSTM, RNN, and DNN approaches. Wu et al. [26] developed a tool wear diagnosis model based on singular value decomposition (SVD) and bidirectional LSTM (BiLSTM). The cutting force signal was taken as the monitoring signal. The current sample period and the previous four samples were taken as input data of the BiLSTM. The results show that the proposed model outperformed other models, such as LSTM, RNN, and bidirectional GRU (BiGRU).

Besides the success of the use of ANNs in the mentioned scenarios, since tool RUL prognosis is gaining more importance, several researchers have been focused on this topic in the recent years. Zhou et al. [27] proposed a method for predicting tool RUL under variable working conditions based on the LSTM network. After extracting the wear characteristics from the process monitoring signals and combining them with the cutting condition variables, they were able to predict the tool RUL. Zheng et al. [22] also proposed a LSTM-based approach for RUL prediction, combining LSTM followed by feed-forward networks. The model was trained with different working conditions on a widely used datasets: milling data set. The results showed that LSTM model outperforms other approaches.

Zhang et al. [28] predicted the RUL of a cutting tool using the Hurst exponent and CNN-LSTM. The data employed for RUL prediction was vibration, cutting force, and AE, and the results showed an accurate prediction. Zhang et al. [29] using the RMS of vibration signals predicted the RUL of a cutting tool of a milling machine. The experiment results showed that the LSTM-based model achieved the best overall accuracy over GRU and Elman RNN models. Yao et al. [30] proposed a deep transfer reinforcement learning (DTRL) network based on LSTM for tool RUL prediction. Force in three directions and vibration in three directions were collected. A 6-level wavelet package decomposition (WPD) was performed, and the results showed that the model was able to predict tool RUL. An et al. [31] proposed a hybrid model combining a CNN with a stacked BiLSTM and LSTM network, named CNN-SBULSTM, to predict the tool RUL in milling process. The results indicated that the employed framework is applicable to track the tool wear evolution and predict its RUL with the average prediction accuracy reaching up to 90%.

Summarizing, as can be seen in the state of the art, RUL is a key topic of research due to the number of studies found recently. ML models are being used more and more due to their importance. Among them, there are very few references of RNN, and there is a wide margin to continue investigating in this field, which can improve the RUL prediction and consequently reduce the costs in the machine tool industry. Thus, the main objectives of this paper are the following:

- First, in order to obtain faster and lighter models and to reduce the complexity and costs related with sensors, data extraction, and computation, optimum features to predict tool RUL are identified using the RF recursive feature elimination (RF-RFE) technique. In the production environment of the machining industry, unlike in the laboratories, there is a limitation in the number of signals that can be captured from the machines. Due to the lack of consensus in the state of the art on what signals are the optimum ones for cutting tool RUL prediction, in this paper, the impact of 26 different signals for the estimation of tool RUL in a turning process is investigated.
- Second, this paper investigates the performance of BiLSTM and BiGRU models for tool RUL prediction in a turning process. It has been proved that BiLSTM performs better than other models for tool wear prediction [26]. However, while tool wear prediction aims to diagnose the condition of the tool, RUL is a prognostics method. To the best of the authors' knowledge, the BiGRU model has never been applied in the field of tool RUL prognostics before. The results obtained for these two RNNs were compared with uni-directional LSTM and GRU, CNNs, MLP, and random forest (RF). Using the proposed methodology, the employed models are trained, optimized, tested, and compared along with the identified optimum features to obtain the lowest RUL prediction error.

2 Materials and methods

This section presents the followed methodology and techniques employed to evaluate the performance of the different models and to identify the optimal predicting signals.

2.1 Methodology

The methodology is depicted in Fig. 1, and it is divided in two general phases. The first one, contained in the blue box, is employed for the investigation of the optimum features for tool RUL prediction. The second one, contained in the gray box, is employed for the investigation of the optimum ML model using the optimum features.

In the first one, data corresponding to 12 different tools was acquired and segmented in N segments of 1 s (where $0 < i \leq N$) (a detailed experimental procedure can be found in Sect. 3). Then, different features (x_j) were extracted from each of the segments and the RUL value (y) of each tool was calculated. Afterwards, using the RF-RFE technique, the best k features were identified for the tool RUL prediction. In the second phase, the dataset was split in two different sets: training and test sets, containing 8 and 4 tools, respectively (randomly selected), as it is a common procedure to split the dataset into 2/3 for training and 1/3 for testing [32]. Both sets were standardized. Then, the training set was used to train and optimize the models. Later, those models were tested using the test set, and the different model results were compared using the RMSE metric. All the described steps are detailed in the following sections.

2.1.1 Signal features extraction

A total of 26 different signals were captured with the purpose of predicting tool RUL. Once recorded, all the signals were split in N segments of 1 s and the following 5 different features were extracted from all of them:

- Mean is the sum of a group of numbers divided by the amount of numbers of that collection:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{1}$$

- RMS is defined as the square root of the mean square (the arithmetic mean of the squares of a set of numbers):

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2} \tag{2}$$

- Maximum is the maximum value of a group of numbers.

$$MAX = \max(x_i) \tag{3}$$

- Skewness is a measure of symmetry in a distribution:

$$Skewness = \frac{\sum_{i=1}^{N}(x_i - \mu)^3}{(N-1)\sigma^3} \tag{4}$$

- Kurtosis is a measure of the "tailedness" of the probability distribution of a variable:

$$Kurtosis = \frac{\sum_{i=1}^{N}(x_i - \mu)^4}{(N-1)\sigma^4} \tag{5}$$

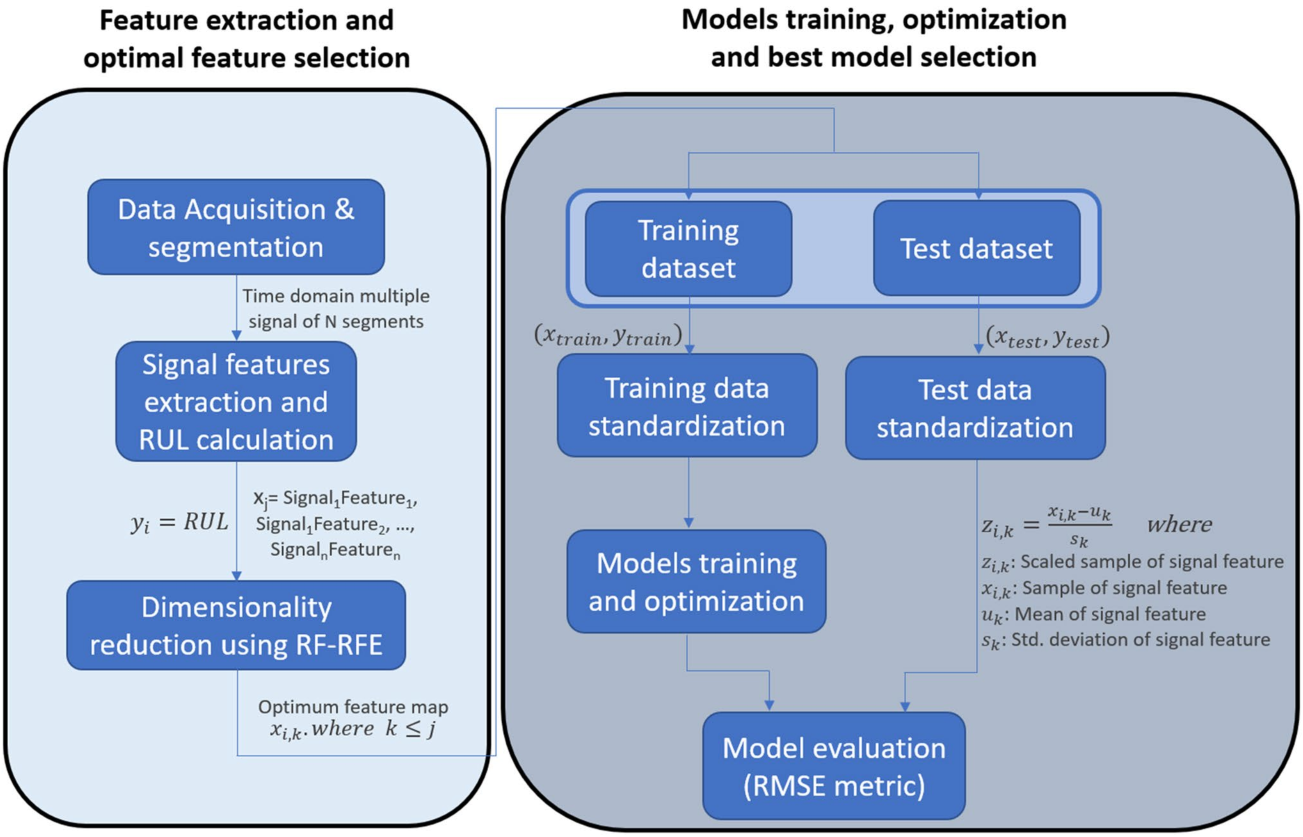

Feature extraction and optimal feature selection
Models training, optimization and best model selection
Data Acquisition & segmentation
Time domain multiple signal of N segments
Signal features extraction and RUL calculation
$y_i = RUL$
X_j= Signal$_1$Feature$_1$, Signal$_1$Feature$_2$, ..., Signal$_n$Feature$_n$
Dimensionality reduction using RF-RFE
Optimum feature map $x_{i,k}.where\ k \leq j$
Training dataset
Test dataset
(x_{train}, y_{train})
(x_{test}, y_{test})
Training data standardization
Test data standardization
Models training and optimization
$z_{i,k} = \frac{x_{i,k}-u_k}{s_k}$ where
$z_{i,k}$: Scaled sample of signal feature
$x_{i,k}$: Sample of signal feature
u_k: Mean of signal feature
s_k: Std. deviation of signal feature
Model evaluation (RMSE metric)

Fig. 1 Graphical description of the followed methodology

After extracting these features to the signals, a total of j variables were created, being x_j each feature extracted from each of the signals. $x_{i,j}$ represents a sample of a feature of a signal.

After obtaining the desired variables, a value of 250 µm was established as the maximum permissible wear of a tool, being a reference of maximum flank wear employed in industry [33]. For that reason, the signals recorded on higher wear values were deleted from the dataset. The RUL value, which is the objective value to be predicted (y), was calculated as the difference between the time when the tool reaches 250 µm wear and the current time.

2.1.2 Dimensionality reduction

Due to the high number of signals captured and the feature extraction process explained in Sect. 2.1.1, a total of 130 predictors were obtained. It is foreseen that not all of them will predict the RUL with the same accuracy and some of them could be correlated, giving the same information. For that reason, to evaluate which of the features explained better the RUL evolution of the tools, RF-RFE algorithm was applied. Thus, correlated and redundant features were eliminated reducing the number of predictors to k. This algorithm, which is explained in detail in Sect. 2.2.2, helps to decrease the training computational demand of ML algorithms and the prediction error.

2.1.3 Data split and standardization

After obtaining the best k variables, before training the models, data was randomly split in two sets: training and test sets, containing 8 and 4 tools, respectively. After this, the data was standardized. Standardization is a common requirement for many ML estimators, as if data does not look like standard normally distributed, they might underperform. The applied operation standardizes features by removing the mean and scaling to unit variance:

$$z = \frac{x - u}{s} \tag{6}$$

where u and s are the mean and standard deviation of the training set, respectively.

2.1.4 Model training and optimization

2.1.4.1 Model training Once the data was split and standardized, BiLSTM and BiGRU models were trained and tested with the features extracted from the RF-RFE algorithm.

Besides, for comparing their performance with other commonly used models, the same procedure was followed with LSTM, GRU, CNN, MLP, and RF with a number of 500 trees.

CNN, MLP, and RF models are fed with a two-dimensional array, represented in Fig. 2:

- Rows: A single record of the features.
- Columns: Different features used to train the models.

However, for the LSTM, BiLSTM, GRU, and BiGRU models, the structure of the input data must be modified. They are fed with a three-dimensional array, represented in Fig. 3:

- Rows: A single record of the features.
- Columns: Different features used to train the models.
- Depth: Number of time-steps used. The time-step value indicates the number of previous samples added to the actual one.

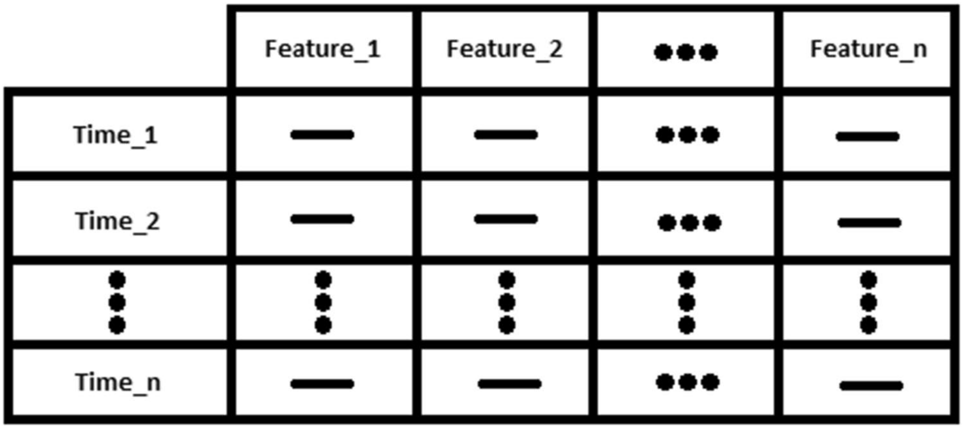
When training these type of models, three parameters must be indicated: the batch size (the number of training examples (rows) in one forward/backward pass), the number of features, and the time-steps used.

To avoid mixing the final and the start of consecutive tools, when training these models, this data structure modification was applied to each of the tools separately, and then all of them were joined again. For RUL prediction, the same idea was followed to avoid mixing the final and start of consecutive tools in the same batch size, and each of the tool RUL was predicted individually.

2.1.4.2 Models' optimization To optimize the RUL prediction, different techniques were applied to the DL models:

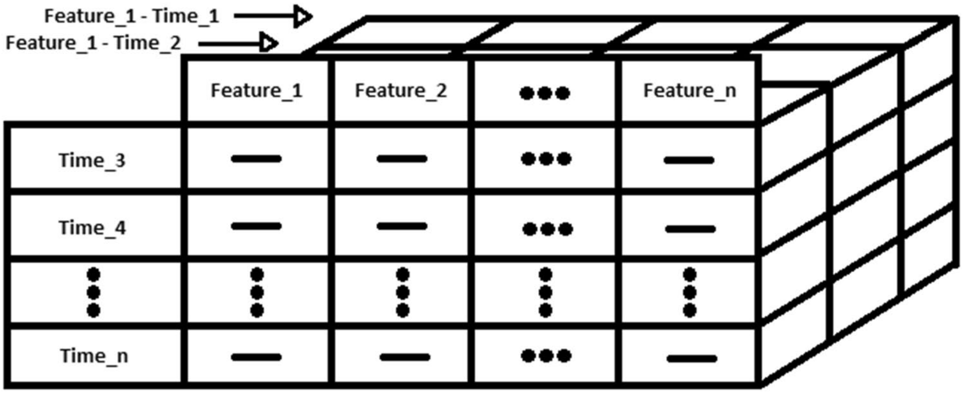
Early stopping: When training the models, the main goal is to minimize the loss function, being in this case the MSE. With the early stopping, the training process is stopped when monitored metric stops improving. That metric is evaluated at the end of each epoch. The patience value, which indicates the number of consecutive epochs in which the loss does not decrease needed to terminate the training process, is a parameter that must be chosen. This was set to 6.

Fig. 2 Data structure to feed traditional ML models



	Feature_1	Feature_2	•••	Feature_n
Time_1	—	—	•••	—
Time_2	—	—	•••	—
⋮	⋮	⋮	⋮	⋮
Time_n	—	—	•••	—

Fig. 3 Data structure to feed LSTM and GRU models



	Feature_1	Feature_2	•••	Feature_n
Time_3	—	—	•••	—
Time_4	—	—	•••	—
⋮	⋮	⋮	⋮	⋮
Time_n	—	—	•••	—

Training process hyperparameter optimization: Using cross-validation (CV), in the training process, all combinations of the predefined hyper parameters fit the selected model (see Table 1). The results are compared and the combination which gives the best results is chosen. This technique is explained in detail in Sect. 3.1. In this work, the dataset was split in training and test sets, the first one containing 8 tools and the second one 4 tools. For the model training, the number of folds of the CV was set to 4, and the following hyperparameters were optimized:

After optimizing the models, they were compared in terms of execution time and RUL prediction error using the RMSE metric. As the procedure from splitting the data until predicting the RUL is stochastic, all the models were trained and test for 100 times taking different tools for training and testing in each iteration. The final execution time and prediction error results were calculated as the mean of the 100 iterations.

2.1.5 *Evaluation metric*

Besides, to obtain a clearer vision of the performance of each individual signal when predicting the RUL of the tools, each of the signals was individually used as input for each of the models. This process was also repeated for 100 times and the mean and standard deviation of the results was calculated.

Table 1 Optimized hyperparameters for DL models

Hyperparameter	Values
Number of layers	From 1 to 4
Number of neurons	From 1 to 5
Batch size [34]	8, 16, 32, 64
Activation function	Rectified linear unit (ReLu), linear, sigmoid and hyperbolic tangent (Tanh)
Learning rate [35] [36]	0.01, 0.05, 0.1

The metric used to evaluate the performance of the models is the RMSE, which is defined as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{N}\left(\hat{y}_i - y_i\right)^2} \quad (7)$$

where $\hat{y}_i$ is the predicted value, y_i the observed values, and n is the observed sample size.

2.2 Materials

In the following sections, a detailed description of the employed techniques in the followed methodology is presented.

2.2.1 *Cross-validation*

In normal ML models, the available data is divided into 2 sets, the training and testing sets. This is a common procedure to ensure that the model can predict previously never seen data. However, in case of wanting to optimize model hyperparameters, this procedure is no longer valid, as there is a risk of overfitting on the test set because the parameters can be tweaked until the estimator performs optimally.

To solve this problem, a great solution to this problem is the CV procedure, which is represented in Fig. 4. With this procedure, a validation set is not any longer needed. Data is partitioned in 2 sets: training and test set. Then, the training set is partitioned in k smaller sets and the following procedure is followed to train the model and choose the best hyperparameters:

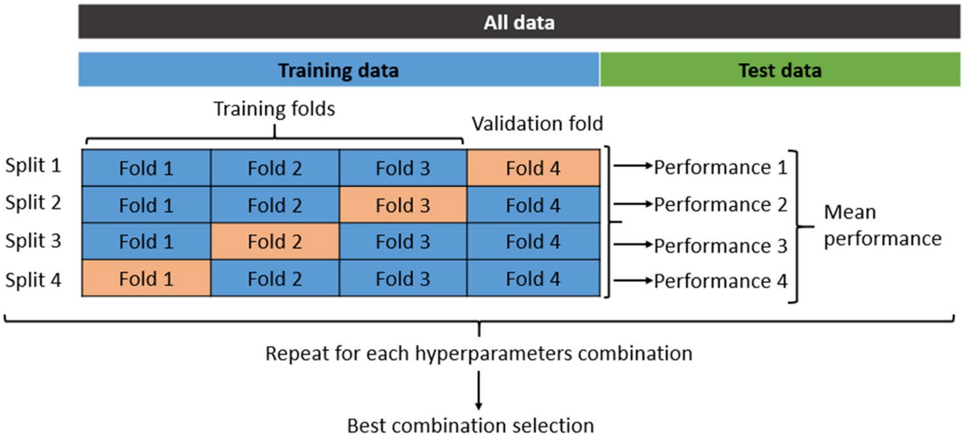


Fig. 4 Hyperparameter selection through CV diagram

- The model is trained using $k-1$ of the sets of the training data.
- The model is evaluated on the remaining set of the data.

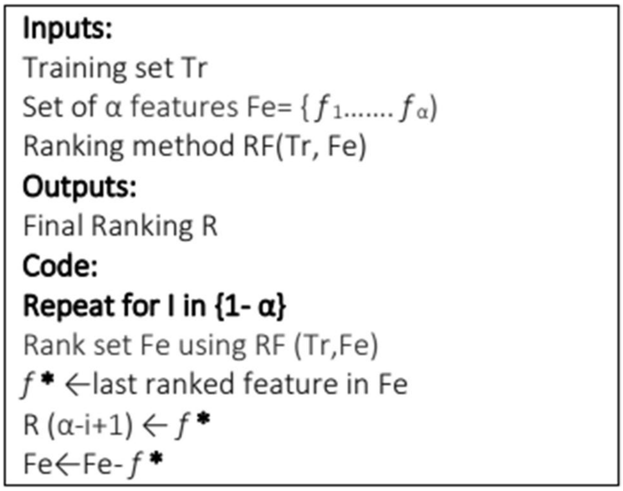
This procedure is repeated k times for each of the possible hyperparameter combinations. This approach is also computationally more expensive.

2.2.2 RF-RFE

High-dimensional data usually reduces the efficiency of the predictive models, as it often contains a lot of redundant and irrelevant information [37]. In order to avoid this drawback, it is necessary to select a subset with most discriminative features. In this study, the redundant and highly correlated set of features generated from the feature extraction of all the sensors is reduced and optimized using the RFE. The RFE technique implements a backward selection of the generated features by ranking their importance to an initial model using all the predictors [38]. It is a greedy optimization procedure used to find the superlative performing subset of features. This technique requires a model to estimate the ranking of the input features. Compared with other models such as SVM or logistic regression, used in [39], RF has been proven to be more effective, being able to use fewer features to get a higher classification accuracy [40]. Thus, RFE based on RF (RF-RFE) is a feature selection method that combines RF to estimate the error for each recursive feature deleted, and RFE, whose process is explained in Fig. 5.

2.2.3 LSTM

This kind of network was especially developed for working with time series, and it is composed of memory cells. LSTM cell is an advanced version of the RNN cell, which solves the vanishing gradient problem [41]. Figure 6 illustrates the basic structure of an LSTM unit/neuron.



```
Inputs:
Training set Tr
Set of α features Fe={f1.......fα}
Ranking method RF(Tr, Fe)
Outputs:
Final Ranking R
Code:
Repeat for I in {1- α}
    Rank set Fe using RF (Tr,Fe)
    f* ←last ranked feature in Fe
    R (α-i+1) ← f*
    Fe←Fe- f*
```

Fig. 5 Pseudo-code of recursive feature elimination process

A LSTM cell is defined as one or more units/neurons corresponding to a time-step. In each time-step, new input data,x_t, is introduced in a cell, as well as the information contained in the units of the previous cell: c_{t-1} and h_{t-1} of the different units. The number of time-steps is defined by the previously explained "depth" user defined parameter. Figure 7 shows the structure of multiple LSTM cells.

Each unit of a cell consists of two main elements (see Fig. 6):

1. The cell state (c): its purpose is to store and transmit memory. It retains an internal state that is not output. The parameter x_t represents the input data at time t, and h_{t-1} is the hidden state at time $t-1.c_{t-1}$ represents the cell state at time $t-1$. This is upgraded to the present cell state (c_t) in the hidden layer at time t.
2. Three gates formed by activation functions and pointwise operations: input gate (i_t), a forget gate (f_t), and an output (o_t) gate. Those are used to protect the c_t. The unit of an LSTM network cell also contains an input node (a_t):

- The a_t is used for updating the c_t, whereas the controlled gates are used to determine whether to allow information to pass through them.
- The f_t determines whether the information of the c_{t-1} should be maintained.
- The i_t determines which f_t's information may pass through it.

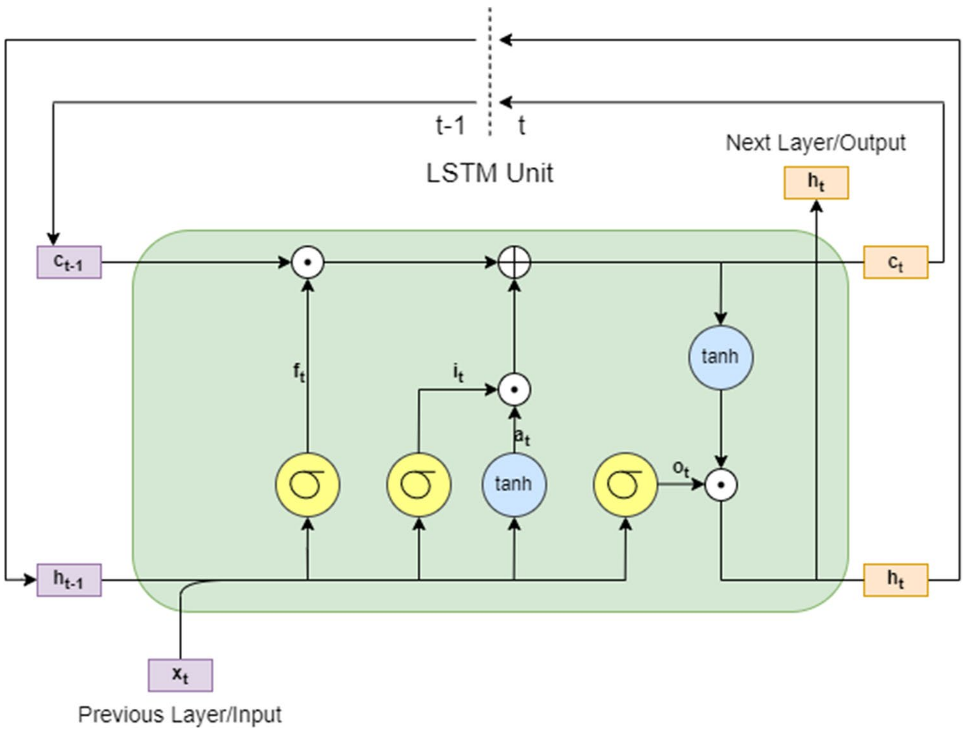
Through Eqs. (8–11), the variables a_t, f_t, i_t, and o_t are calculated.

Once the selected information is passed through the i_t, the c_t is calculated by doing an element-wise addition with the vectors (information) passing through the f_t. The calculation of this process is expressed in Eq. (12). The o_t determines which c_t's information may pass through it. The vectors (information) passing through the o_t, which are the output vectors of the current hidden layer, are the hidden state at time t (h_t). The calculation method for h_t is presented in Eq. (13). In addition, the c_t and h_t are transmitted to the hidden layer at time $t+1$. This process that progresses with the time series is used for the transmission and learning of memory.

$$a_t = \tanh\left(W_a x_t + H_a h_{t-1} + b_a\right) \tag{8}$$

$$f_t = \sigma\left(W_f x_t + H_f h_{t-1} + b_f\right) \tag{9}$$

Fig. 6 Basic structure of LSTM unit



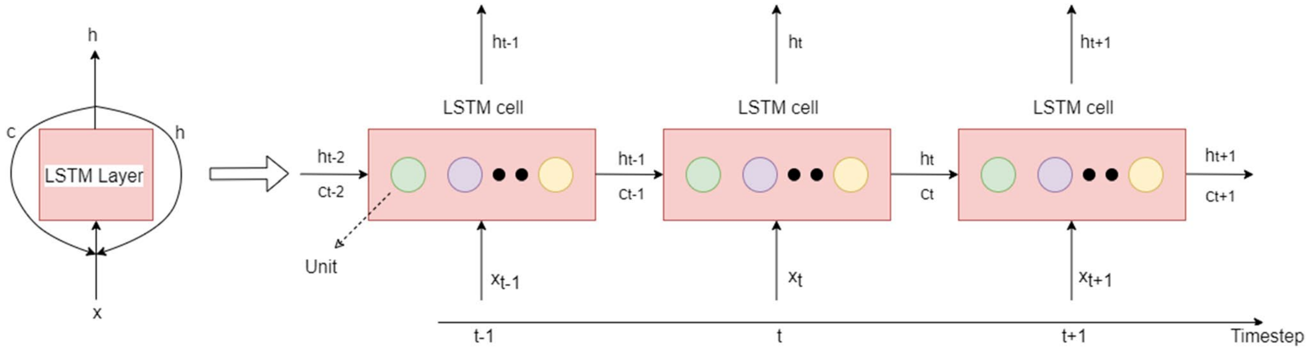


Fig. 7 Representation of LSTM layer both in folded and unfolded forms

$$i_t = \sigma\left(W_i x_t + H_i h_{t-1} + b_i\right) \tag{10}$$

$$o_t = \sigma\left(W_o x_t + H_o h_{t-1} + b_o\right) \tag{11}$$

$$c_t = \left(f_t \odot c_{t-1}\right) \odot \left(i_t \odot a_t\right) \tag{12}$$

$$h_t = o_t \odot \tanh(c_t) \tag{13}$$

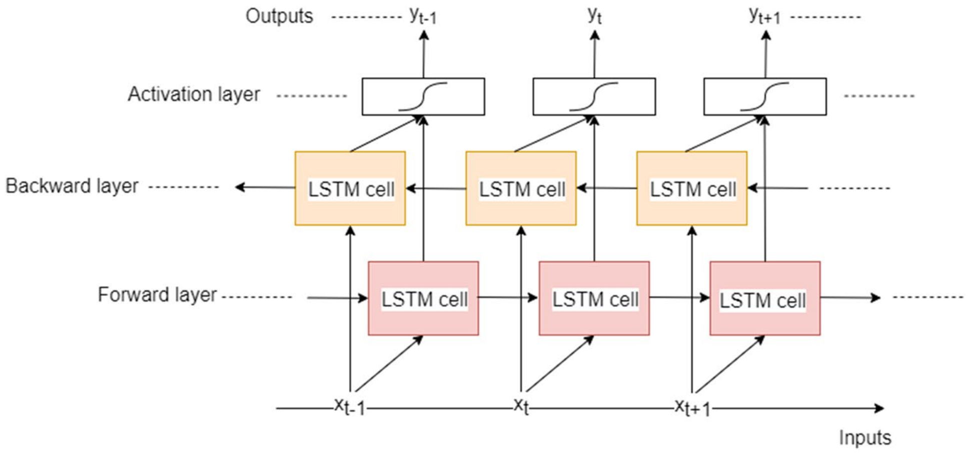
Where W and H represent the weight, b denotes the bias, $\oplus$ is the symbol for element wise addition, $\odot$ is the symbol for element-wise multiplication, and tanh and σ represent the hyperbolic tangent and the sigmoid activation functions, respectively.

As explained above, unidirectional LSTM only preserves information of the past because the only inputs it has seen are from the past. However, by using BiLSTM, it will run the inputs in two ways, one from past to future and one from future to past (see Fig. 8). Like this, through the layer that runs backwards, the information from the future is preserved. This means that the model can preserve information from both past and future in any time by using the two hidden states combined.

2.2.4 GRU

This kind of network is also composed of memory cells, being a small variant of the LSTM cell. It was proposed by [42] and the GRU unit is simpler than the LSTM unit, having less

Fig. 8 BiLSTM 1-layer NN



gates and thus fewer weights to be tuned (see Fig. 9). Instead of having 3 gates, it contains just 2: reset gate (r) and update gate (z). The main difference between GRU and LSTM is that GRU does not possess an internal memory (c_t), applying the r directly to the h_{t-1}. The r determines how to combine new input x_t with that of the previous memory in its cells. On the other hand, the z acts like the f_t of LSTM and decides what information to throw out and what information to add.

The equations for the GRU cell are similar to the LSTM ones:

$$z_t = \sigma\left(W_z x_t + U_z h_{t-1} + b_z\right) \tag{14}$$

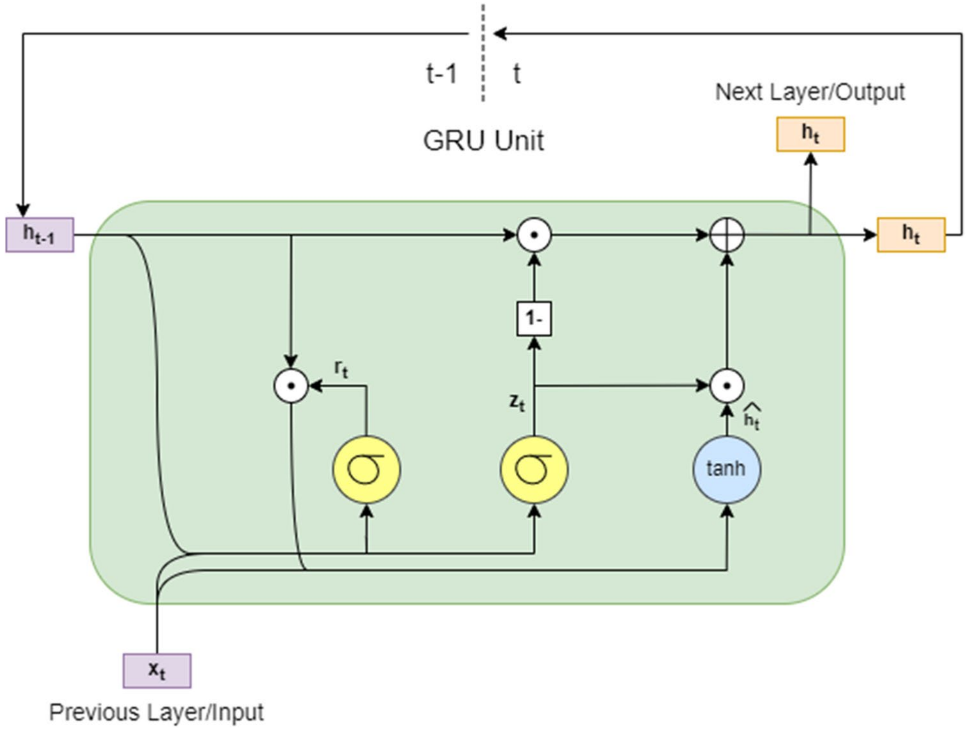
$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{15}$$

$$\hat{h}_t = \tanh\left(W_h x_t + U_h\left(r_t \odot h_{t-1}\right) + b_h\right) \tag{16}$$

$$h_t = \left(1 - z_t\right) \odot h_{t-1} + z_t \odot \hat{h}_t \tag{17}$$

where W and U represent the weight, b denotes the bias, $\odot$ is the symbol for element-wise multiplication, and tanh and σ represent the hyperbolic tangent and the sigmoid activation functions, respectively.

Fig. 9 Basic structure of a GRU unit



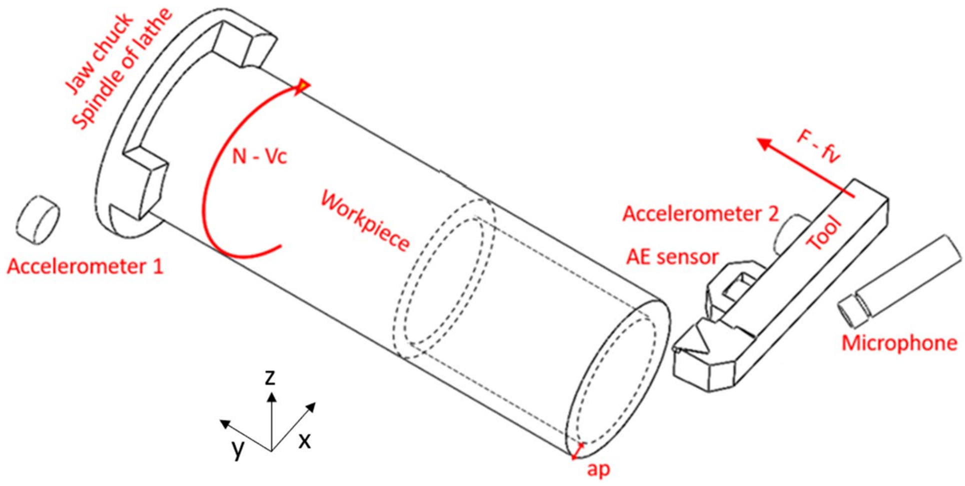


Fig. 10 Schematic representation of the turning process and location of sensors

Table 2 Equipment used and recorded parameters

Sensor	Equipment	Parameter	Range	Freq	Signals
PCB356A16 triaxial accelerometer 1	NI9234	Vibrations	±5 V	50 kHz	$\boldsymbol{A}_{x1}$ $\boldsymbol{A}_{y1}$ $\boldsymbol{A}_{z1}$
PCB356A16 triaxial accelerometer 2	NI9234	Vibrations	±5 V	50 kHz	$\boldsymbol{A}_{x2}$ $\boldsymbol{A}_{y2}$ $\boldsymbol{A}_{z2}$
B&K 4189-A-021 microphone	NI9234	Sound	±5 V	50 kHz	***Sound***
Kistler 9121	NI9239	Machining force in *x*/*y*/*z*-axis	±10 V	50 kHz	$\boldsymbol{F}_x$**(Radial force)** $\boldsymbol{F}_y$ **(Feed force)** $\boldsymbol{F}_z$ **(Tangential force)**
Machine currents	NI9227	Motor currents in *y*-axis	±5 Arms	50 kHz	$\boldsymbol{I}_{uy}$ $\boldsymbol{I}_{vy}$
	NI9227	Currents of spindle motor	±5 Arms	50 kHz	$\boldsymbol{I}_{us}$ $\boldsymbol{I}_{vs}$
Machine voltages	NI9225	Motor voltage in *y*-axis	300 Vrms	50 kHz	$\boldsymbol{V}_{uy}$ $\boldsymbol{V}_{vy}$ $\boldsymbol{V}_{wy}$
	NI-9244	Voltage of spindle motor	400 Vrms	50 kHz	$\boldsymbol{V}_{us}$ $\boldsymbol{V}_{vs}$ $\boldsymbol{V}_{ws}$
Kistler 8152B acoustic emissions	NI9223	Vibrations, high frequency	±10 V	1 MHz	***AE***
Mechanical power CNC	CNC	Mechanical power of spindle motor	±2,147,483,647 W	250 Hz	$\boldsymbol{TV}50_s$
Electrical power CNC	CNC	Electrical power of spindle motor	±100,000 W	250 Hz	$\boldsymbol{TV}51_s$
Torque CNC	CNC	Torque of spindle motor	±1000% of the stall torque of the motor N	250 Hz	$\boldsymbol{TV}2_s$
	CNC	Motor torque in *x*-axis	±1000% of the stall torque of the motor N	250 Hz	$\boldsymbol{TV}2_x$
	CNC	Motor torque in *y*-axis	±1000% of the stall torque of the motor N	250 Hz	$\boldsymbol{TV}2_y$

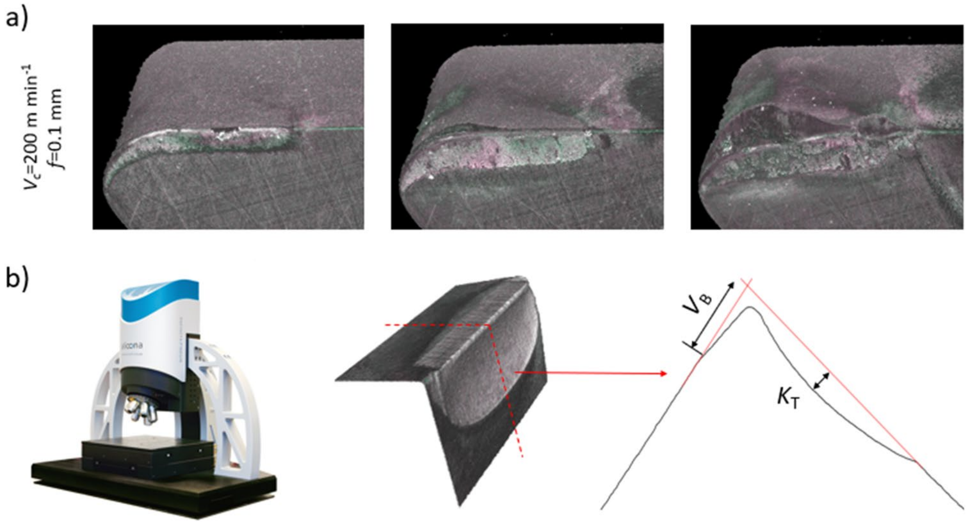


Fig. 11 Wear measurement strategy with Alicona Infinite Focus G4

This kind of networks can train faster since there are only few parameters in the U and W matrices and hence may be useful when not much training data is available [29].

3 Experimental procedure

Due to the high variability that wear has in machining operations, the number of repetitions carried out was 12, to develop predictive models with high degree of confidence. All the trials were done at fixed cutting conditions, with a cutting speed (*Vc*) of 200 m/min, a feed rate (*fv*) of 0.1 mm/rev and a depth of cut (*ap*) of 2 mm.

The material employed for the tests was a 19NiMoCr6 steel, which has a totally bainitic microstructure. Concerning the cutting tools, P25 grade uncoated inserts were employed, reference Widia TPUN160308TTM. These were clamped to a Widia CTGPL2020K16 tool holder, which gives an effective rake and clearance angles of 5° and 6°, respectively, with a positioning angle of 90°.

Table 3 Number of wear measurements for each tool

Tool	Number of tool wear measurements
1	4
2	3
3	6
4	7
5	7
6	8
7	6
8	9
9	7
10	6
11	7
12	5

3.1 Measured variables and employed equipment

The tool holder was clamped to a Kistler 9121 dynamometer to record the cutting forces. The acquisition frequency was set to 50 kHz, using a National Instrument cDAQ-9178 with an analog input module NI-9239.

Two triaxial accelerometers PCB356A16 were employed to record the vibrations during the turning process. One was clamped to the lathe close to the location of the spindle (accelerometer 1) and the other to the tool holder (accelerometer 2). The acquisition frequency was set to 50 kHz, using a National Instrument cDAQ-9178 with an analog input module NI-9234.

A microphone B&K 4189-A-021 was also used to measure the sound signals. The acquisition frequency was set to 50 kHz, using a National Instrument cDAQ-9178 with an analog input module NI-9234.

The acoustic emissions were recorded with a Kistler 8152B sensor coupled with a Type 5125B conditioning system. This was magnetically attached to the tool holder, as shown in Fig. 10. The acquisition frequency was set to 1 MHz, using a National Instrument cDAQ-9178 with an analog input module NI-9223.

In addition to the abovementioned equipment, the current and voltage signals during cutting were also measured. Both were recorded in the motor of the *y*-axis drive and the motor of the spindle. The spindle motor currents were measured with LEM ITB 300-S transducers. The acquisition frequency was set to 50 kHz, using a National Instrument cDAQ-9178

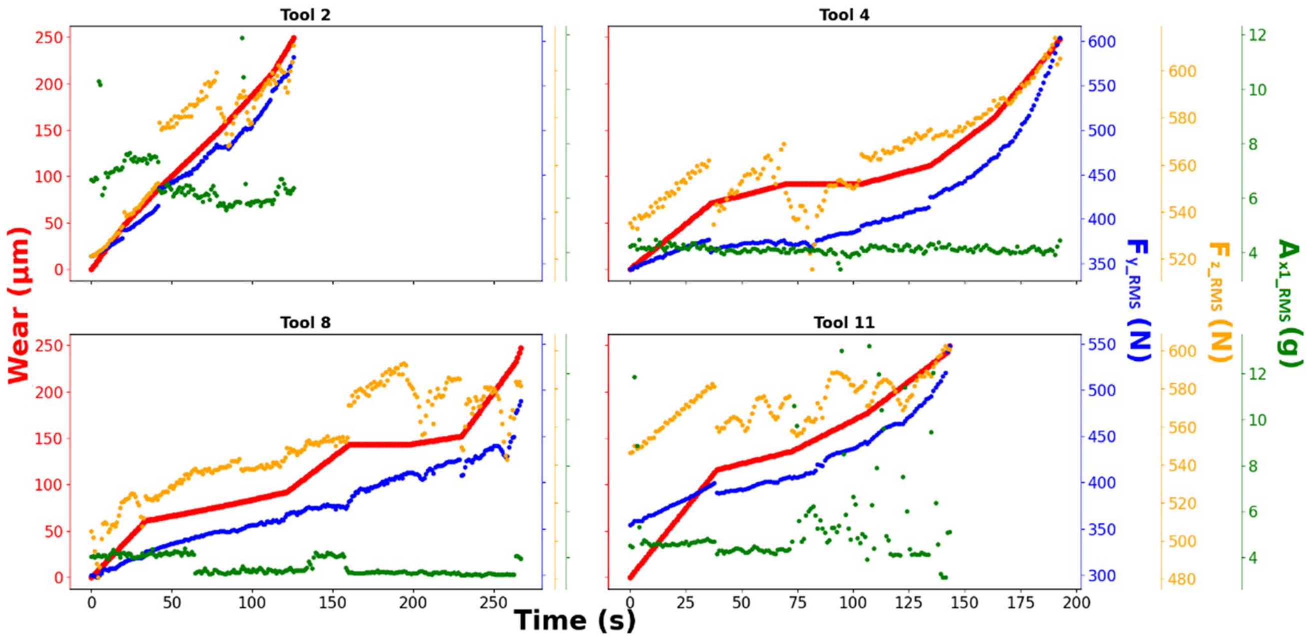


Fig. 12 Wear VS F_{y_RMS}, F_{z_RMS} and A_{x1_RMS} features evolution over time of tools 2, 4, 8, and 10

with analog input modules NI-9225 and NI-9244 for voltages and NI-9227 for currents.

Table 2 summarizes the measured variables and the equipment employed.

3.2 Cutting procedure

The cutting procedure was as follows:

1. Machining of a predefined length of the workpiece, commonly 1/3 of the available length (70 mm).
2. Cleaning of the tool insert to remove adhered material and to enable a correct measurement of tool wear.
3. Tool wear measurement using an Alicona Infinite Focus G4 profilometer. This profilometer permits the 3D measurement of the wear in the flank (flank wear *Vb*) and rake faces (crater wear *Kt*).
4. Restart the process (1)–(3) until wear in the flank face (*Vb*) exceeds a value of 250 µm, being a reference of maximum flank wear employed in industry.

The following Fig. 11a shows 3D data sets of the evolution of tool wear of one of the repetitions, obtained with the Alicona Infinite Focus G4. To establish *Vb*, from each captured 3D geometry of the worn tool, the profile of the mid-plane of the contact section was extracted. The localization and an example of a profile is shown in Fig. 11b. The measured wear mode, *Vb*, is specified in the profile.

Table 3 shows the number of wear measurements carried out for each of the tools. In each tool, to get the wear between measurements, a linear interpolation has been performed. It is worth noting that the number of tool wear measurements changes considerably. This is due to the stochastic nature of the process. Thus, although all the parameters of the cutting process are the same, the duration of the tool until the wear reaches 250 µm wear varies considerably and consequently the number of wear measurements.

4 Results

In order to gain a first insight of the dataset and to ensure that the wear measurements of each tool were correctly done, Fig. 12 shows the evolution over time of the tool wear (in red) along with the evolution of three different features extracted from the signals of 4 different tools. Each of the points where the red line gradient changes indicates that in that time the tool wear has been measured. Data between

Table 4 Cumulative rank of the 3 best features obtained by the RF-RFE algorithm (run for 10 times)

Features	Cumulative rank
F_{y_RMS}	16
F_{y_MEAN}	19
F_{y_MAX}	28

measurements follow a straight line since a linear interpolation has been performed to calculate the values between measurements. The F_{y_RMS} feature (blue) presents a significant correlation with tool wear, which suggest that it may be a good predictor of RUL. The RMS feature of the cutting force signal (F_{z_RMS}) (yellow) also seems to follow the increasing trend. However, the RMS feature of the x-axis head accelerometer (A_{x1_RMS}) (green) does not show a direct correlation with the wear, indicating that this feature alone may not be a good RUL predictor. It is worth mentioning that even though the different tools perform and are measured under the same working conditions, due to the stochastic nature of the process, each of them wears down differently and has different number of wear measurements.

To evaluate which features are the best predictors of RUL, RF-RFE algorithm was applied. This algorithm concluded that the best combination for RUL prediction was just the F_{y_RMS} feature. Table 4 shows the rank of the 3 best features obtained by the RF-RFE technique, evaluating them individually. As it is a stochastic algorithm, it was run for 10 times, and the results show the sum of the ranks obtained in each iteration. The results show that the F_y signal was the optimum one. As expected, the features F_{y_RMS} and F_{y_MEAN} obtain remarkably similar results, as the RMS and mean metrics are highly correlated. In 8 iterations, these 2 features were identified as the optimum predictors by RF-RFE algorithm.

Then, 5 different ML models were trained and optimized with the F_{y_RMS} feature. Subsequently, to compare the performance of each signal predicting RUL, a model was trained and tested for 100 times using the features extracted from each of the signals as input predictor. In addition, all models

Table 5 RMSE and standard deviation test results of different models, trained separately with each of the signals (mean of 100 calculations)

		MODELS				
		MLP	LSTM	GRU	CNN	RAND.FORST
SIGNALS	$A E$	63.42 +/- 12.32	72.01 +/- 13.32	67.27 +/- 13.64	64.54 +/- 11.75	75.89 +/- 11.12
	A_{x1}	67.12 +/- 14.39	78.39 +/- 11.33	74.88 +/- 12.44	66.92 +/- 12.58	78.94 +/- 9
	A_{x2}	67.33 +/- 13.3	78.21 +/- 29.01	72.6 +/- 15.3	66.73 +/- 11.76	77.01 +/- 10.14
	A_{y1}	68.42 +/- 12.85	76.54 +/- 12.44	73.94 +/- 13.97	67.57 +/- 11.92	77.78 +/- 6.95
	A_{y2}	71.36 +/- 39.44	73 +/- 46	74 +/- 32	66.59 +/- 18.46	72.63 +/- 11.37
	A_{z1}	73.12 +/- 20.04	77.02 +/- 13.42	73.86 +/- 13.79	69.79 +/- 17.06	82.54 +/- 9.11
	A_{z2}	69.28 +/- 12.92	79.08 +/- 35.6	74.05 +/- 20.85	67.43 +/- 11.26	74.45 +/- 7.63
	F_x	55.76 +/- 9.95	71.95 +/- 14.55	65.08 +/- 12.18	56.01 +/- 9.05	66.6 +/- 12.86
	F_y	32.34 +/- 8.75	36.18 +/- 12.25	33.33 +/- 8.61	32.72 +/- 8.37	38.97 +/- 11.51
	F_z	44.11 +/- 9.83	52.81 +/- 10.07	47.98 +/- 10.71	44.14 +/- 9.94	48.41 +/- 8.36
	I_{us}	70.66 +/- 12.27	81.66 +/- 13.66	75.7 +/- 12.92	69.1 +/- 11.24	71.49 +/- 15.5
	I_{uy}	73.67 +/- 19.12	74.92 +/- 18.32	73.57 +/- 15.77	74.62 +/- 19.57	75.87 +/- 12.59
	I_{vs}	73.1 +/- 13.37	88.64 +/- 20.07	77.53 +/- 15.02	73.86 +/- 11.59	76.7 +/- 16.06
	I_{vy}	71.58 +/- 22.92	72.98 +/- 28.92	69.28 +/- 16.01	66 +/- 10.59	66.59 +/- 12.56
	Sound	97.86 +/- 225.59	74.08 +/- 64.65	65.35 +/- 16.93	67.11 +/- 14.15	71.79 +/- 10.29
	V_{us}	68.87 +/- 12.37	85.06 +/- 14.71	78.66 +/- 12.66	70.07 +/- 11.67	80.94 +/- 7.81
	V_{uy}	74.27 +/- 10.49	73.98 +/- 10.99	71.34 +/- 12.39	72.48 +/- 10.05	75.67 +/- 9.76
	V_{vs}	70.56 +/- 12.4	83.23 +/- 15.4	79.16 +/- 14	70.57 +/- 11.66	79.45 +/- 8.91
	V_{vy}	72.5 +/- 12.56	77.58 +/- 12.98	73.64 +/- 12.81	70.79 +/- 11.07	75.06 +/- 11.66
	V_{ws}	69.08 +/- 11.57	83.48 +/- 13.95	77.1 +/- 12.41	70.43 +/- 11.77	80.5 +/- 8.89
	V_{wy}	70.84 +/- 13.89	74.94 +/- 13.74	69.73 +/- 13.89	68.88 +/- 10.4	74.63 +/- 10.96
	$TV50_s$	68.98 +/- 16.85	70.9 +/- 13.43	68.01 +/- 14.82	67.44 +/- 12.81	74.07 +/- 14.06
	$TV51_s$	67.65 +/- 14.08	72.3 +/- 16.05	75.35 +/- 60.34	67.94 +/- 14.24	69.9 +/- 11.48
	$TV2_s$	64.38 +/- 11.92	76.03 +/- 12.99	68.81 +/- 11.67	64.65 +/- 11.1	68.46 +/- 11.1
	$TV2_x$	66.51 +/- 12.15	76.39 +/- 13.81	70.22 +/- 14.07	65.93 +/- 12.04	82.65 +/- 11.49
	$TV2_y$	58.63 +/- 14.77	63.55 +/- 13.1	59.2 +/- 14.43	59.21 +/- 13.28	66.7 +/- 10.58
	RFE (F_{y_RMS})	30.98 +/- 8.35	32.91 +/- 12.42	31.12 +/- 8.35	33.45 +/- 11.57	38.74 +/- 9.9

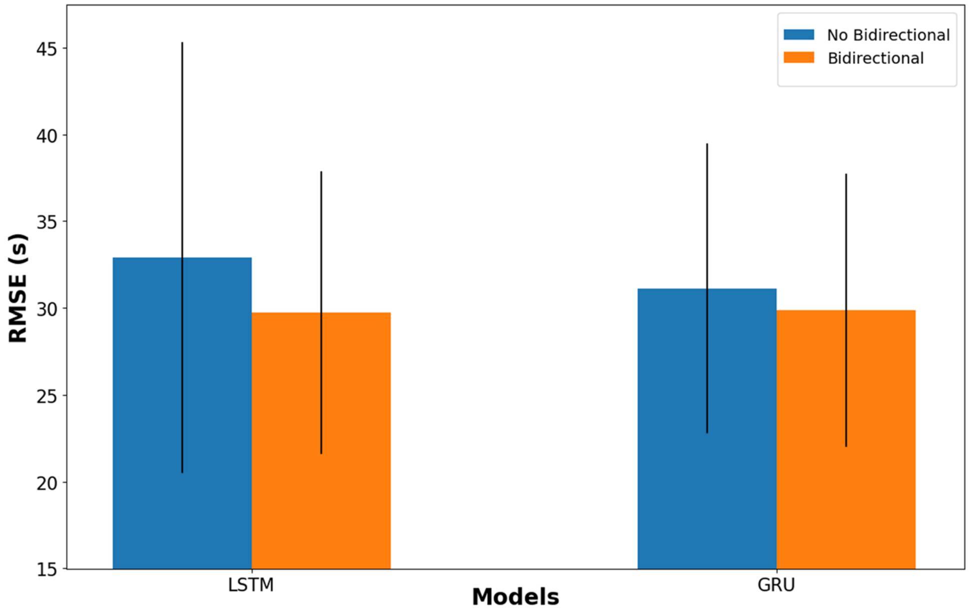


Fig. 13 RMSE and standard deviation test results of LSTM and GRU models with and without bidirectional "property" (mean of 100 calculations)

Table 6 Optimized hyperparameters of the 3 best models trained and tested with F_{Y_RMS} feature

Models	Hidden layers	Neurons	Batch size	Epoch	Activation functions	Optimizer	Time steps	Learn. rate
MLP	1	4	16	100 max (patience: 6)	Hidden layer: ReLu Output layer: linear	Adam	–	0.1
BiLSTM	1	3	16	100 max. (patience: 6)	Hidden layer: ReLu Output layer: linear	Adam	5	0.01
BiGRU	1	2	32	100 max. (patience: 6)	Hidden layer: linear Output layer: linear	Adam	5	0.05

were tested and trained 100 times using F_{y_RMS}, as it was estimated by RF-RFE algorithm as the optimum predictor. Table 5 displays all the testing mean RMSE results and standard deviation of all the models along with the different signals. The RMSE results of the table contain a conditional color scale. The lowest RMSE results are colored in green while the highest ones are shown in red. Best results were obtained from the F_y signal, especially just using the F_{y_RMS} feature, estimated by the RF-RFE algorithm as the best predictor. As expected from Fig. 12, the F_z signal also

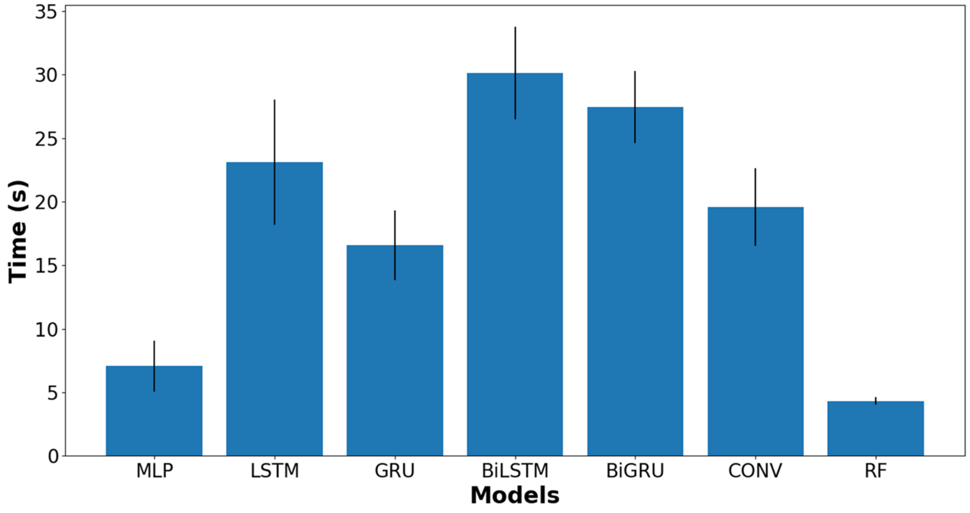


Fig. 14 Training execution time and standard deviation results of different models (mean of 100 calculations)

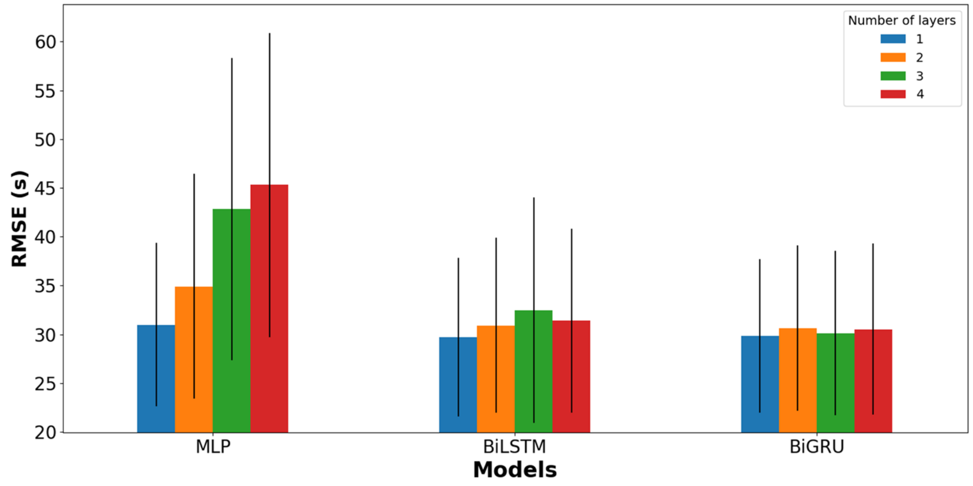


Fig. 15 RMSE and standard deviation test results of MLP, BiLSTM, and BiGRU models with different number of hidden layers (mean of 100 calculations)

achieved positive results. However, other widespread signals employed in the machinery industry, such as AE and current signals, showed high RMSE. For example, the obtained RMSE using AE features as predictors was roughly 100% higher than the RMSE result of the RUL calculated with the F_{y_RMS} feature. Looking at the optimum feature (F_{y_RMS}), LSTM, GRU, and MLP, trained with 1 hidden layer, were the models that incurred in the lowest RMSE, being the last one the best one. With some variables, the values reported in LSTM, GRU, and MLP are not much better than the ones obtained with CNN or RF. However, when comparing the best scenarios, the differences are high. MLP improves by 7.76 s (19.52%) and 2.47 s (7.38%) the RMSE of RF and CNN models using the F_{y_RMS} feature.

Figure 13 compares the mean test RMSE and standard deviation of training the RNN models, with 1 hidden layer, with and without using the bidirectional property for 100 times with the F_{y_RMS} feature. When training the LSTM and GRU models with the bidirectional property, the results improved significantly. As expected, as the bidirectional models run the inputs in two ways, one from past to future and one from future to past; they were able to preserve information from both past and future in any time and improve the RMSE results. Thus, the BiLSTM and BiGRU models, compared with LSTM and GRU, reduced the RMSE by 3.18 s (9.66%) and 1.26 s (4.04%), respectively, providing the lowest RMSE of 29.73 and 29.86 s, respectively, of all models investigated. Moreover, BiLSTM and BiGRU models, comparing with MLP, improved the RMSE by 1.25 s (4.03%) and 1.12 s (3.61%), respectively, using the F_{y_RMS} feature.

Table 6 shows the optimized hyperparameters of the best 3 models trained with the F_{y_RMS} feature. As just one variable was used as input parameter, simple models were obtained. The best performance was obtained using just one hidden layer, which implies a reduction of the computational requirements of the training of the models.

Figure 14 displays the mean training execution time and standard deviation of the different ML models using the F_{y_RMS} feature as predictor. As DL models are more complex than traditional ML ones, the firsts took more time to train. As expected, it can also be observed that using the bidirectional property, models also took longer to train, as

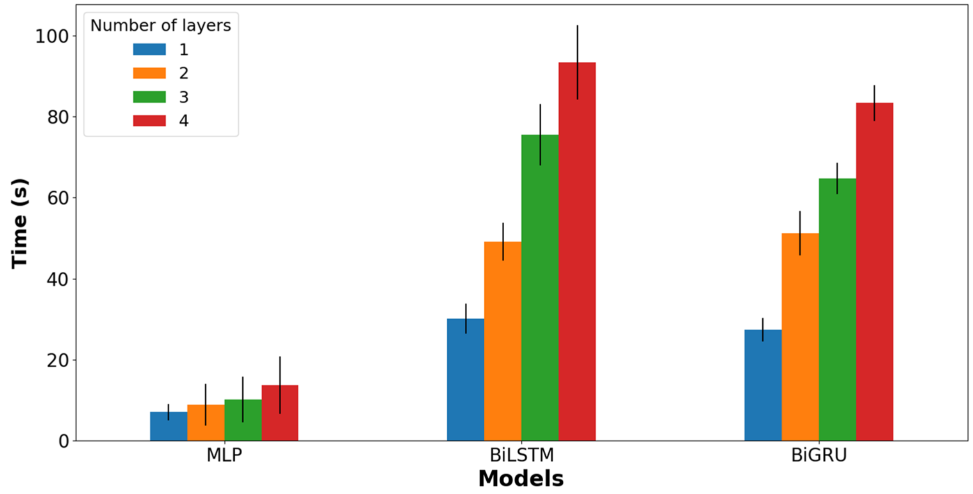


Fig. 16 Training execution time and standard deviation results of MLP, BiLSTM, and BiGRU models with different number of hidden layers (mean of 100 calculations)

they execute the inputs in two ways, one from past to future and one from future to past. The model which needed more time to be trained was the BiLSTM. Due to the simplicity of the BiGRU model compared to BiLSTM, its training execution time was slightly lower. On the other hand, the model that trained faster was the RF one, being 6.94 times faster than the BiLSTM, closely followed by the MLP.

Figure 15 displays the RMSE obtained with MLP, BiGRU, and BiLSTM models using different number of hidden layers. It reveals that in the three models, increasing the number of layers, in general, had a negative impact in the RMSE obtained. This fact especially affected to the MLP model, being the RMSE of BiLSTM and BiGRU models less sensitive to the addition of hidden layers, being the RMSE obtained using one hidden layer 30.98, 29.73, and 29.86 s, respectively.

Figure 16 shows the impact of the number of hidden layers in the training time. As expected, a greater number of hidden layers resulted in slower training models, as they had to adjust more parameters. This phenomenon especially affects BiLSTM and BiGRU models, being the execution time of those models 3.04 and 3.03 times higher with 4 layers compared with just 1 layer, respectively.

Figure 17 shows the real versus predicted RUL using BiGRU and BiLSTM models using F_{y_RMS} as predictor. The results show that in both models in initial stages, the models' prediction error is higher. However, those predictions improve as the RUL of the tools reach to their end. The final stages of the tool RUL are more critical than the early ones, as an incorrect prediction in last stages could lead to change the tool too early and not optimize its RUL. Moreover, it could lead to change it too late and increase the probability of manufacturing a defective piece. As mentioned before, these models obtain better results than traditional models. For example, the RUL prediction RMSE of BiLSTM and BiGRU models, comparing with the next model that obtain better results, MLP, improves by 1.25 s (4.03%) and 1.12 s (3.61%), respectively, using the F_{y_RMS} feature. BiLSTM and BiGRU models perform similarly. For that reason, in scenarios where the computational optimization is required, the BiGRU model could be optimum. Otherwise, if the model predictive performance is the main goal, BiLSTM model could be the most appropriate one.

5 Conclusions

In the machine tool sector, one of the main aspects is to monitor the wear of the cutting tools, as it affects directly to the fulfillment of tolerances, production of scrap, energy

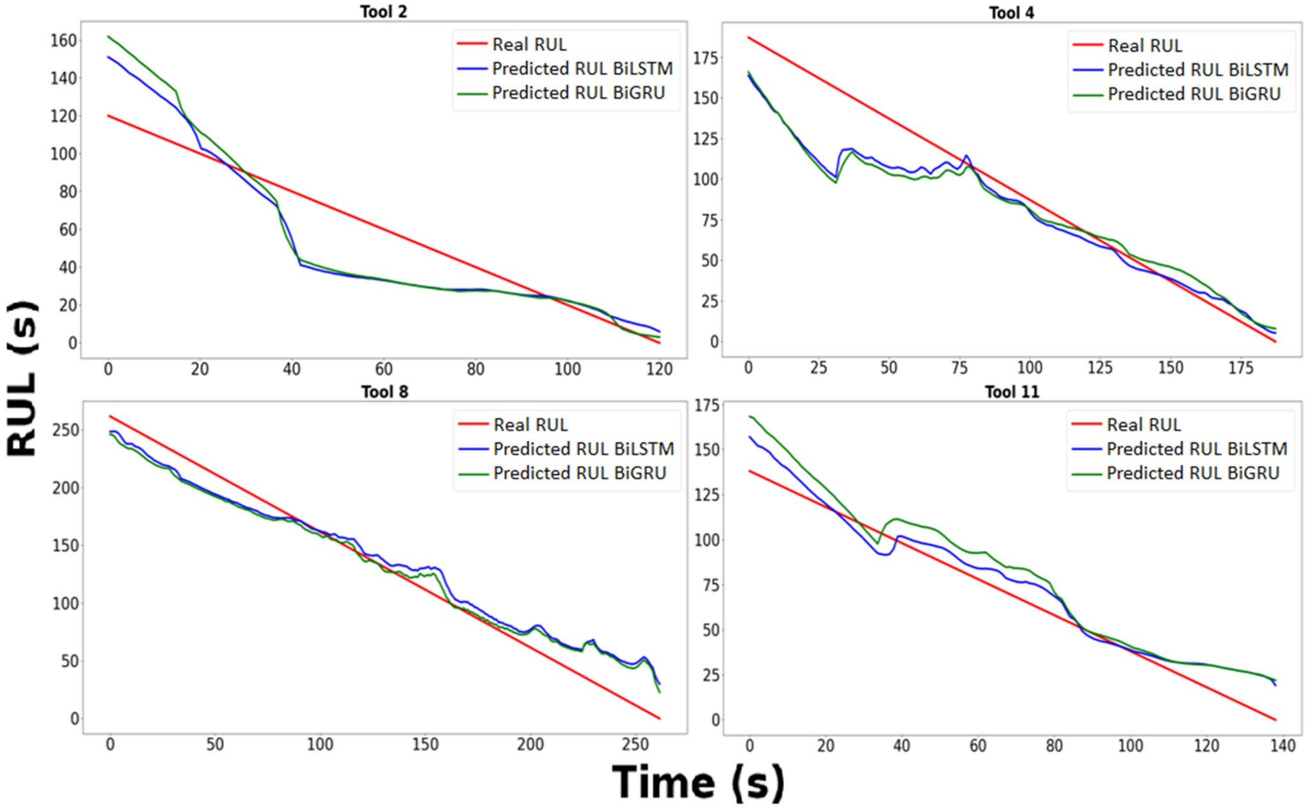


Fig. 17 Real vs predicted RUL of tools 2, 4, 8, and 11 with BiLSTM and BiGRU models

consumption, etc. Besides, the prediction of the remaining useful life (RUL) of the cutting tools, which is related to their wear level, is gaining more importance in the field of predictive maintenance, being that prediction is a crucial point for an improvement of the quality of the cutting process. Thus, in this paper, the prediction of tool RUL by using BiRNNs was investigated. The models were evaluated using multi-sensor data captured from a turning process. In the production environment of the machining industry, unlike in the laboratories where a great and different number of signals can be measured, there is a limitation in the number of signals that can be captured from the machines. For that reason and unlike in other works, RF-RFE technique was applied to the features extracted from the signals to select the best input features of the models. According to the results obtained using RF-RFE technique, among all the features extracted, the F_{y_RMS} is the best predictor of tool RUL, which corresponds to the RMS value of feed force. This means that the RUL of the tools can be predicted with just one feature, reducing substantially the complexity and costs of the data acquisition system to just one sensor. Besides, the complexity signal pre-processing and the ML and DL models is also reduced, achieving accurate and faster models.

Different regressive ML and DL models, including RF, MLP, LSTM, GRU, BiLSTM, BiGRU, and CNN, were trained and optimized with the F_{y_RMS} feature, identified by the RF-RFE algorithm as the optimum predictor. According to what is seen in the literature in tool wear [26] and RUL [31] prediction, BiGRU and BiLSTM models outperform other trained models such as LSTM, GRU, MLP, RF, and CNN. When applying the bidirectional property to LSTM and GRU models, their RUL prediction RMSE is improved by 3.18 s (9.66%) and 1.26 s (4.04%), respectively. Also, the BiGRU model, which is a simpler BiLSTM variant, performs 8.95% faster than BiLSTM when training. For that reason, in scenarios where the computational optimization is required, the BiGRU model could be optimum. Otherwise, if the model predictive performance is the main goal, BiLSTM model could be the most appropriate one.

Future extension of this work may include the study of data from performing the turning operation with variable cutting conditions or different type of machining processes such as milling and drilling, to assure that the F_{y_RMS} is the optimum feature for tool RUL prediction. In addition, custom loss metrics could be used to train the models, giving more importance to the predictions when the tools get closer to their end of life and penalizing more the predictions that surpass the real RUL. Finally, effort could be put into combining DL approaches with other data-driven or physics-based approaches. These hybrid approaches have a great potential and could provide more accurate and robust RUL prediction.

Author contribution Conceptualization: TFDB, JLF. Data curation: JLF, TFDB. Formal analysis: TFDB, JLF. Investigation: XB, MSDB, JV. Methodology: TFDB, JLF. Project administration: JLF. Resources: XB, MSDB, JV. Software: TFDB. Supervision: AG. Visualization: TFDB. Writing—original draft: TFDB. Writing—review and editing: all authors.

Funding This research was partially funded by the Department of Industry of the Basque Government within Elkartek program (KK-2020/00103) and by the Basque Government (SPRI) within the Elkartek programme (KK-2021/00111 ERTZEAN).

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Gouriveau R and Ramasso E (2010) From real data to Remaining Useful Life estimation: an approach combining neuro-fuzzy predictions and evidential Markovian classifications, *38th ESReDA Seminar Advanced Maintenance Modelling*, no. October 2010, pp. 1–13
2. Kurada S, Bradley C (1997) A review of machine vision sensors for tool condition monitoring. Comput Ind 34(1):55–72. https://doi.org/10.1016/s0166-3615(96)00075-9
3. Li B (2012) A review of tool wear estimation using theoretical analysis and numerical simulation technologies. Int J Refract Metals Hard Mater 35:143–151. https://doi.org/10.1016/j.ijrmhm.2012.05.006
4. Ferrando Chacón JL, Fernández de Barrena T, García A, Sáez de Buruaga M, Badiola X, Vicente J (2021) A novel machine learning-based methodology for tool wear prediction using acoustic emission signals. Sensors (Basel) 21(17):1–16. https://doi.org/10.3390/s21175984
5. Duspara M, Sabo K, Stoić A (2014) Acoustic emission as tool wear monitoring. Tehnicki Vjesnik 21(5):1097–1101
6. Wang Y, Zhao Y, Addepalli S (2020) Remaining useful life prediction using deep learning approaches: a review. Procedia Manuf 49(2019):81–88. https://doi.org/10.1016/j.promfg.2020.06.015
7. Cubillo A, Perinpanayagam S, Esperon-Miguez M (2016) A review of physics-based models in prognostics: application to gears and bearings of rotating machinery. Adv Mech Eng 8(8):1–21. https://doi.org/10.1177/1687814016664660
8. Li X, Ding Q, Sun JQ (2018) Remaining useful life estimation in prognostics using deep convolution neural networks. Reliab Eng Syst Saf 172:1–11. https://doi.org/10.1016/j.ress.2017.11.021

9. Elattar HM, Elminir HK, Riad AM (2016) Prognostics: a literature review. Complex Intell Syst 2(2):125–154. https://doi.org/10.1007/s40747-016-0019-3
10. Heng A, Zhang S, Tan ACC, Mathew J (2009) Rotating machinery prognostics: state of the art, challenges and opportunities. Mech Syst Signal Process 23(3):724–739. https://doi.org/10.1016/j.ymssp.2008.06.009
11. Byrne G, Dornfeld D, Inasaki I, Ketteler G, König W, Teti R (1995) Tool condition monitoring (TCM) - the status of research and industrial application. CIRP Ann Manuf Technol 44(2):541–567. https://doi.org/10.1016/S0007-8506(07)60503-4
12. Diei EN, Dornfeld DA (1987) Acoustic emission from the face milling process–the effects of process variables. J Manuf Sci E T ASME 109(2):92–99. https://doi.org/10.1115/1.3187114
13. Muñoz-Escalona P, Díaz N, Cassier Z (2012) Prediction of tool wear mechanisms in face milling AISI 1045 steel. J Mater Eng Perform 21(6):797–808. https://doi.org/10.1007/s11665-011-9964-6
14. Ma M, Sun C, Chen X (2017) Discriminative deep belief networks with ant colony optimization for health status assessment of machine. IEEE Trans Instrum Meas 66(12):3115–3125. https://doi.org/10.1109/TIM.2017.2735661
15. Schwabacher MA (2005) A survey of data-driven prognostics. Collection of Technical Papers - InfoTech at Aerospace: Advancing Contemporary Aerospace Technologies and Their Integration 2(May):887–891. https://doi.org/10.2514/6.2005-7002
16. Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition YANN. Proceedings of the IEEE 86(11):2278–2324. https://doi.org/10.1109/5.726791
17. Wang J, Ma Y, Zhang L, Gao RX, Wu D (2018) Deep learning for smart manufacturing: methods and applications. J Manuf Syst 48:144–156. https://doi.org/10.1016/j.jmsy.2018.01.003
18. Malhi A, Yan R, Gao RX (2011) Prognosis of defect propagation based on recurrent neural networks. IEEE Trans Instrum Meas 60(3):703–711. https://doi.org/10.1109/TIM.2010.2078296
19. Elman JL (1990) Finding structure in time. Cogn Sci 14(2):179–211. https://doi.org/10.1207/s15516709cog1402_1
20. Heimes FO (2008) Recurrent neural networks for remaining useful life estimation, *2008 International Conference on Prognostics and Health Management, PHM 2008*, https://doi.org/10.1109/PHM.2008.4711422
21. Bengio Y, Simard P, Frasconi P (1994) Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Netw 5(2):157–166. https://doi.org/10.1109/72.279181
22. Zheng S, Ristovski K, Farahat A, and Gupta C (2017) Long short-term memory network for remaining useful life estimation, *2017 IEEE International Conference on Prognostics and Health Management, ICPHM 2017*, pp. 88–95, https://doi.org/10.1109/ICPHM.2017.7998311
23. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
24. Huang R, Xi L, Li X, Richard Liu C, Qiu H, Lee J (2007) Residual life predictions for ball bearings based on self-organizing map and back propagation neural network methods. Mech Syst Signal Process 21(1):193–207. https://doi.org/10.1016/j.ymssp.2005.11.008
25. Tian Z (2012) An artificial neural network method for remaining useful life prediction of equipment subject to condition monitoring. J Intell Manuf 23(2):227–237. https://doi.org/10.1007/s10845-009-0356-9
26. Wu X, Li J, Jin Y, Zheng S (2020) Modeling and analysis of tool wear prediction based on SVD and BiLSTM. Int J Adv Manuf Technol 106(9–10):4391–4399. https://doi.org/10.1007/s00170-019-04916-3
27. Zhou JT, Zhao X, Gao J (2019) Tool remaining useful life prediction method based on LSTM under variable working conditions. Int J Adv Manuf Technol 104(9–12):4715–4726. https://doi.org/10.1007/s00170-019-04349-y
28. Zhang X, Lu X, Li W, Wang S (2021) Prediction of the remaining useful life of cutting tool using the Hurst exponent and CNN-LSTM. Int J Adv Manuf Technol 112(7–8):2277–2299. https://doi.org/10.1007/s00170-020-06447-8
29. Zhang J, Zeng Y, Starly B (2021) Recurrent neural networks with long term temporal dependencies in machine tool wear diagnosis and prognosis. SN Appl Sci 3(4):1–13. https://doi.org/10.1007/s42452-021-04427-5
30. Yao J, Lu B, Zhang J (2022) Tool remaining useful life prediction using deep transfer reinforcement learning based on long short-term memory networks. Int J Adv Manuf Technol 118(3–4):1077–1086. https://doi.org/10.1007/s00170-021-07950-2
31. An Q, Tao Z, Xu X, El Mansori M, Chen M (2020) A data-driven model for milling tool remaining useful life prediction with convolutional and stacked LSTM network. Measurement (Lond) 154:107461. https://doi.org/10.1016/j.measurement.2019.107461
32. Nguyen QH et al. (2021) Influence of data splitting on performance of machine learning models in prediction of shear strength of soil, Math Probl Eng, vol. 2021, https://doi.org/10.1155/2021/4832864
33. Tool-life testing with single-point turning tools (1993) *ISO 3685:1993*
34. Cao X, Chen B, Yao B, and Zhuang S (2019) An intelligent milling toolwear monitoring methodology based on convolutional neural network with derived wavelet frames coefficient, Appl Sci (Switzerland), 9(18) https://doi.org/10.3390/app9183912
35. Goodfellow I, Bengio Y, and Courville A (2016) *Deep learning*. MIT Press
36. Zeiler MD (2012) ADADELTA: an adaptive learning rate method. http://arxiv.org/abs/1212.5701
37. Wu Y and Zhang A (2004) Feature selection for classifying high-dimensional numerical data, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, https://doi.org/10.1109/cvpr.2004.1315171
38. Guyon I, Weston J, Barnhill S (2002) Gene selection for cancer classification using support vector machines. Mach Learn 46:389–422. https://doi.org/10.1023/A:1012487302797
39. Deshpande P, Pandiyan V, Meylan B, Wasmer K (2021) Acoustic emission and machine learning based classification of wear generated using a pin-on-disc tribometer equipped with a digital holographic microscope. Wear 476(xxxx):203622. https://doi.org/10.1016/j.wear.2021.203622
40. Chen Q, Meng Z, Liu X, Jin Q and Su R (2018) Decision variants for the automatic determination of optimal feature subset in RF-RFE, Genes (Basel), 9(6), https://doi.org/10.3390/genes9060301
41. Huang YC and Chen YH (2021) Use of long short-term memory for remaining useful life and degradation assessment prediction of dental air turbine handpiece in milling process, Sensors, 21(15) https://doi.org/10.3390/s21154978
42. Chung J, Gulcehre C, Cho K and Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling, NIPS 2014 Workshop on Deep Learning