Received 1 June 2025, accepted 25 July 2025, date of publication 1 August 2025, date of current version 7 August 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3595058

RESEARCH ARTICLE

Comprehensive Evaluation of Techniques for Intelligent Chatter Detection in Micro-Milling Processes

GUILHERME SERPA SESTITO[1], **WESLEY ANGELINO DE SOUZA**[1], **(Member, IEEE), ALESSANDRO ROGER RODRIGUES**[2], **AND MAÍRA MARTINS DA SILVA**[2]

[1]Department of Electrical Engineering, Federal University of Technology—Paraná (UTFPR), Cornélio Procópio, Paraná 86300-013, Brazil
[2]Department of Mechanical Engineering, University of São Paulo (USP), São Carlos, São Paulo 13566-590, Brazil

Corresponding author: Guilherme Serpa Sestito (sestito@utfpr.edu.br)

This work was supported in part by Brazilian Funding Agency through National Council for Scientific and Technological Development (CNPq) under Grant 303884/2021-5 and in part by the São Paulo Research Foundation (FAPESP) under Grant 2019/00343-1.

ABSTRACT Frictional and regenerative chatter are the most relevant unstable dynamic conditions in micro-milling. Both phenomena promote chatter, affecting the surface finishing and reducing the tool life. Therefore, in-process chatter detection strategies are of utmost importance to avoid adverse effects on productivity. These strategies can use machine learning (ML) and deep learning (DL) classifiers, but they should be able to lead with tight computational requirements. A smaller yet relevant set of features could aid these classifiers in coping with these requirements. This work proposed using feature selection to evaluate the impact of several statistical features on the performance of ML classifiers for chatter detection during micro-milling operations, compare them to the performance of the Convolutional Neural Network algorithm, and discuss the employability of the techniques on the STM32F446RE microcontroller. This study exploits experimental data from chatter and chatter-free cuts during machining operations with commercial COSAR-60 low-carbon steel under two different grain sizes (as received and ultra-fined). Random Forest, Pearson's Correlation, and collinearity analysis are the strategies used to identify the most relevant features in these operations. The performance of several ML classifiers is compared in each feature reduction stage with the Deep Learning algorithm. The results discuss the need for applying complex algorithms since the accuracy and F1-score indices presented similar values regardless of the number of features. Moreover, reducing the dataset dimension significantly decreased training/testing time, and mainly in execution latency (FLOPs). This reduction is essential for implementing strategies for in-process chatter detection.

INDEX TERMS Micro-milling, feature selection, correlation analysis, convolutional neural network, machine learning classifiers, chatter identification.

I. INTRODUCTION

Conventional and micro-milling processes employ similar machine components, tool geometry, and cutting fluids; however, the material removal mechanisms differ. For example, the shearing forces that act on the rake face are the main material removal mechanisms of the conventional milling process. In contrast, the plowing and shearing regimes are present in the micro-milling operations [1]. These complex material removal dynamics yield instabilities generated by tool deflections, tool runout, and machining chatter.

Frictional and regenerative chatter are the most relevant unstable dynamic conditions in micro-milling. The former occurs when specific spindle speeds and friction conditions between the cutter and the material lead to an accentuated increase in vibration [2]. The latter happens due to a chip thickness variation in a specific combination of process parameters. This chip variation results from the modulations left on the surface during successive cuts, generating self-exciting vibrations as illustrated in Figure 1 [3].

The associate editor coordinating the review of this manuscript and approving it for publication was Bing Li.

Both phenomena yield large forces and displacements that promote chatter, affect surface finish, and reduce tool life [4], [5]. Therefore, researchers have been investigating strategies to identify stable and unstable conditions to avoid the adverse effects of chatter on productivity. Stability Lobe Diagrams that differentiate the stable and unstable conditions have been extensively used. These diagrams can be computed by time-domain data obtained from experimental campaigns [6], [7] or numerical analysis of dynamic models [8], [9]. However, the derivation of the diagrams may be cumbersome given the complexity of the phenomenon due to the diversity of factors that can influence its appearance, such as the cutting tool, the tool holder, the workpiece material, the machine structure, and the cutting parameters [10].

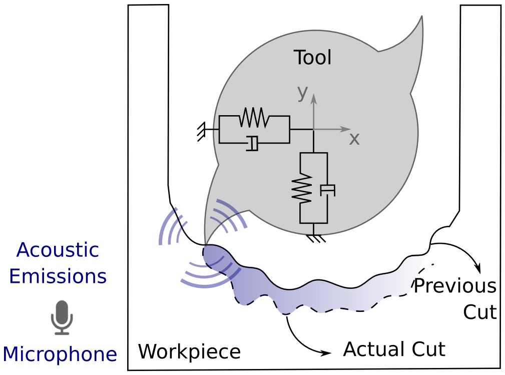


FIGURE 1. Illustration of chatter monitoring using acoustic emission during a micro-milling operation.

As stated in [11], advanced real-time data processing and decision-making methods are still necessary for milling operations. Reference [12] pointed out that chatter detection strategies require signal acquisition, feature extraction, and an indicator of chatter occurrence. Regarding signal acquisition, [13] mentioned that most works explore vibration signals from acquired accelerometers or microphones. However, other signals, such as the motor current/torque and data processed from captured images, can also be exploited [14]. Regarding micro-milling, Acoustic emissions (AE), measured by microphones as illustrated in Figure 2, are the most common data used in chatter detection. Due to the dimensions of the process, the vast majority of sensors may modify the system dynamics, which is not the case with microphones. This low-cost measurement alternative is non-invasive, and its signals contain information on vibrations, tool-workpiece contact, surface integrity, and topography [15], [16], [17]. On the other hand, it is also well known that acoustic signals are susceptible to noise [18], [19] that requires data processing.

These data processing techniques can be applied in the time and/or frequency domain. Real-time data processing and decision-making methods may benefit from data described in the time domain, in which features can be extracted directly from raw, resampled, or filtered signals [13]. The choice of chatter indicator can vary between extracting information from the data (such as entropy, for example), using machine learning techniques, and multi-signal analysis [5]. Due to this necessity and the difficulties in dynamic modeling and measuring relevant data for chatter detection, machine learning strategies have played an essential role in the recent advances in this field. For example, [20] proposed using wavelet transfer and deep convolutional neural networks (NN) to detect chatter in milling operations using the data obtained by a low-cost and non-invasive accelerometer. They achieved 82 to 100% accuracy according to the scenario. Higher accuracy was achieved by [21], where the authors combined k-NN with data processing techniques and obtained 98%. The same performance was obtained by [22] when adopting the Adaboost Algorithm and comparing its result with the SVM (96.7%), Random Forest (96.7%), 1D Convolutional Neural Network (61.7%), and Multilayer Perceptron (90%). Reference [10] applied different classifiers, obtaining satisfactory results, with the accuracy of Perceptron and SVM-RBF close to 100%. Unlike the cited works, [23] exploited machine learning in feature selection using fuzzy entropy. The results obtained by [23] demonstrate the importance of a proper feature selection to improve a classifier's performance.

It is a fact that several machine learning techniques have been proposed in the related literature to detect chatter. However, an extensive discussion is needed regarding the applicability of a method or a set of techniques, from the preprocessing of the dataset, feature selection, classification, and the computational effort for its actual implementation on a microcontroller.

In this sense, this paper compares the applicability of machine learning algorithms (K-NN, SVM, DT, MLP, AdaBoost) and a deep learning algorithm (1D-CNN) for chatter classification in two different materials. The discussion is carried out by analyzing performance indicators, namely training and testing time, F1-Score, and execution latency measured in Floating Point Operations per Second (FLOP), considering the implementation of the algorithms on the STM32F446RE microcontroller. Feature selection algorithms will be applied to the datasets. Proper selection is of utmost importance for the performance of intelligent classifiers.

In this sense, this paper explores feature selection techniques to select, manipulate, and transform raw data into features that can be used in a rule-based algorithm for chatter detection in micro-milling operations. This work extracts 45 features from AE signals obtained during micro-milling operations. These signals contain samples of operations with and without chatter. We propose evaluating these features in three stages, exploring the Random Forest algorithm, Pearson correlation, and collinearity analysis. The outcome of this proposal is the selection of the most relevant features for training machine learning-based classifiers. We train and compare six different classifiers to assess the reduction in processing time, improvement in their performance, and the impact on computational efforts. Determining which feature is most suitable for chatter detection allows the use of simpler classifiers.

The main contributions of this work can be understood as follows:

- This paper extensively discusses which machine learning technique is most appropriate to be applied in chatter classification in micro-milling processes. Both Machine Learning and Deep Learning algorithms are considered.
- To support the choice of the most suitable technique for the chatter classification problem, performance indicators such as accuracy, training, and testing times are considered. Additionally, and differently from the related literature, the computational effort is assessed through FLOPs, assuming the methodology is implemented on the STM32F446RE microcontroller. This brings practical and real meaning to the discussion presented.
- A critical contribution is the feature selection process presented. A set of rules can be applied through a single feature extracted from the acoustic signal, a technique with reduced computational effort. Complex algorithms such as Deep Learning are not always necessary.

This article is organized as follows. We describe the experimental campaign and the data acquired during micro-milling operations in Section II. The feature selection techniques and the machine learning-based classifiers are summarized in Sections III and IV. Section V details the proposal of this work, while Section VI presents and discusses the results. Conclusions are drawn in Section VII.

II. EXPERIMENTAL CAMPAIGN AND ACQUIRED DATA

Micro-milling machining experiments were performed using a slow cutting strategy in a Romi D800 CNC milling machine with a 1 µm position accuracy. This cutting strategy can manufacture micro-channels in the workpiece with the tool diameter. Therefore, the analysis did not consider the tool's flexibility and runout. Figure 2 depicts the workpiece (8 × 26×60 mm), the tool holder, and the microphone (AE sensor).

The workpiece material plays an essential role in the chatter occurrence. Feature selection may reveal that the most relevant features may not be the same for different materials because of differences in their microstructure. Therefore, we investigated two materials: the "as received" COSAR-60 biphasic low-carbon steel (ferrite-pearlite) with 11 µm grain size named AR, and the ultra-fined grain COSAR-60 monophasic (ferrite) with 0.7 µm grain size termed UF.

The AE signals were acquired while cutting four microchannels using a 1mm diameter carbide endmill tool with two flutes. The channels are 26mm long and 1mm wide. The cutting parameters were set to 125 m/min cutting speed, 3µm/tooth feed, 100 µm depth of cutting, and 1.0 mm width. A piezoelectric microphone with a dynamic response of up to 1 MHz captured the AE signals. These signals were acquired and processed on an NI PCI-6251 board at a rate of 1.25 MHz. A high-pass filter with a 250 Hz cutoff and an amplification of 35 dB was used during this acquisition.

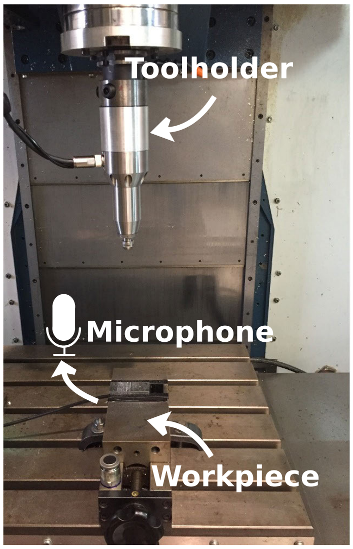


FIGURE 2. Micro-milling machining experiments.

These signals acquired during micromilling operations are shown in [10] and [16]. For the sake of illustration, Figure 3 illustrates the AE signals during the cutting process of the fourth micro-channel of the AR workpiece. The signals during chatter-free and chatter cuts are shown in this figure.

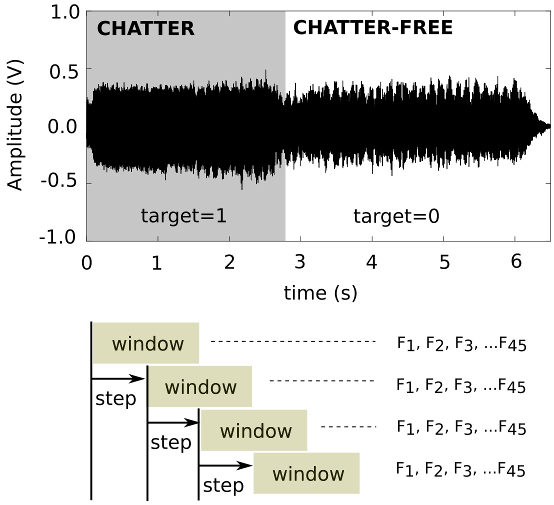


FIGURE 3. Time domain AE signal from micro-milling of AR in the micro-channel 4, an illustration of the features' extraction using sliding window.

III. DATA ANALYSIS AND FEATURE EXTRACTION

According to [13], data analysis is a fundamental step for correctly and efficiently detecting chatter during machining operations. Therefore, this work proposes to analyze raw data in the time domain by extracting and evaluating the features. Feature selection approaches have been described by [10], [16], [24], and [25]; however, a comprehensive evaluation using feature selection is still lacking.

Table 1 describes the features exploited in the work and indicates the related published works in which these features have also been investigated for chatter detection. This proposal only considers statistical features extracted from the data packets for versatility. A sliding window algorithm is responsible for splitting the data into data packets that contain w samples. This algorithm spams the entire dataset moving a window of w samples (a data packet) by s samples (as illustrated in Figure 3). The sliding window algorithm is a widely used strategy, as pointed out by [26].

TABLE 1. Set of extracted features.

Number (j)	Feature (F_j)	Related works
F_1	Average of signal amplitude	[10]
F_2	Standard deviation of signal amplitude	[10, 27]
F_3	Mean deviation	
F_4	Variance of signal amplitude	[10, 12, 25]
F_5	Coefficient of variation of signal amplitude	[5, 10]
F_6	Median	
F_7	RMS value of signal amplitude	[10, 12, 25]
F_8	Kurtose	[12, 25, 27]
F_9	Mode	
F_{10}	Highest absolute value	[10]
F_{11}	Range	
F_{12}	First quartile	
F_{13}	Second quartile	
F_{14}	Third quartile	
F_{15}-F_{45}	Histogram	

Reference [28] point out that a feature selection algorithm promotes insight into data, improves the classifier model, and increases generalization by identifying relevant and irrelevant features. Furthermore, the authors emphasize that this identification may reduce the computational efforts due to a reduction in the dataset dimensionality. This work investigates three feature-selection algorithms for reducing the dataset dimensionality by comparing the classifier's performance and computational effort.

Firstly, we employed the Random Forest (RF) algorithm to reduce the 45 features described in Table 1 for classifying the occurrence of chatter. In Figure 4, $target = 0$ means that the dataset comes from a chatter-free cut; otherwise ($target = 1$), from a cut under the chatter occurrence. In the literature, authors, such as [29] and [30], used this algorithm for classification, regression, and feature selection since it generates ensemble decision trees, as illustrated in Figure 4, considering random subsets of features. The RF algorithm derives multiple decision trees, and their average can be taken by votes that result in the output classifier [31]. In this work, we derive 100 decision trees for performing this average.

Moreover, the algorithm optimally divides a node using the Gini impurity measure, used to identify the most relevant features for the classifications, $F_?$ in Figure 4, and the decision value for the true/false division, *Value* in Figure 4. For generality, the subindex ? indicates that any feature can be used. A decision tree can have fewer nodes than the features presented, revealing the most relevant features for classification. For example, this strategy successfully selected the features for fire detection in videos as reported in [32] and identified individual loads in electrical energy consumption as proposed by [33].

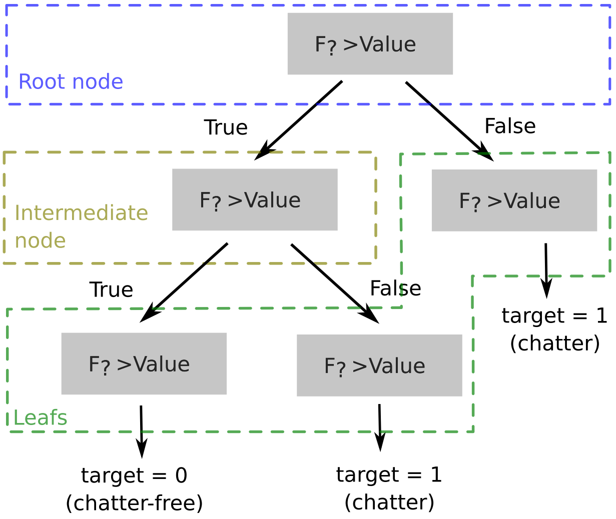
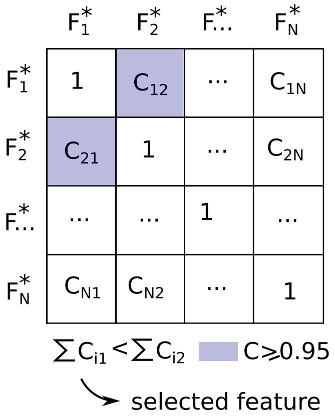


FIGURE 4. Illustration of a Decision Tree for Classification of Chatter.

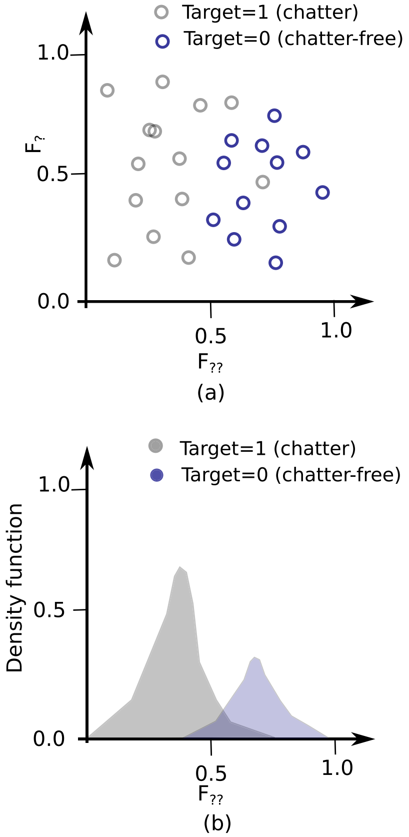
Secondly, we evaluate the correlation between the N most relevant features selected by RF, denoted as F_i^* where $i = 1 \ldots N$, using Pearson correlation as a metric. The idea is to eliminate correlated features since they represent redundant information [34]. References [35] and [36] exploited this metric to reduce the number of features used by the classifiers to detect real-time Ethernet traffic events and forecast electrical demands.



	F_1^*	F_2^*	$F_{\ldots}^*$	F_N^*
F_1^*	1	C_{12}	…	C_{1N}
F_2^*	C_{21}	1	…	C_{2N}
$F_{\ldots}^*$	…	…	1	…
F_N^*	C_{N1}	C_{N2}	…	1

FIGURE 5. Illustration of the Correlation Matrix and the methodology for selecting the most relevant features.

In the present work, we build a matrix of Pearson's correlation between two features, as illustrated in Figure 5. For example, $C_{12} = C_{21}$ is the Pearson correlation of the two most relevant features identified by the RF algorithm. We propose eliminating one of the features if its Pearson correlation is greater than 0.95. This high value indicates that the evaluated features are redundant. The elimination criterion is based on the sum of all Pearson correlation values regarding the redundant features. The feature with the highest sum of Pearson correlation values is eliminated. Figure 5 illustrates this procedure. In this figure, F_1^* and F_2^* are considered redundant since $C_{12} = C_{21} \leq 0.95$. In this way, one of them should be eliminated by evaluating the sum of their Pearson correlation values. In this illustration, the feature F_2^* is eliminated since $\sum_i^N C_{i2} > \sum_i^N C_{i1}$. This procedure reduces the number of relevant features to $M < N$.



(a)

(b)

FIGURE 6. Illustration of the two kinds of plots used in the collinearity analysis.

Finally, a collinearity analysis is performed to select the most important feature among the chosen ones in the study based on the values of Pearson's correlation. This analysis has been done by plotting pairwise relationships in a dataset as [33]. The features chosen using the Pearson correlation metric are used to generate these plots. Figure 6 shows two types of these plots derived from this strategy. Figure 6(a) shows the target values for two features to illustrate the separability of these features regarding the targets. In contrast, Figure 6(b) illustrates the kernel density estimation, an estimator of the probability density function of a random variable, for a determined feature to demonstrate if the targets' distribution is placed in distinct feature values.

IV. CLASSIFIERS

This work investigates the above-mentioned feature-selection algorithms for reducing the dataset dimensionality. This investigation compares five-well-known machine learning-based classifiers: k-nearest neighbours (k-NN) [37], support vector machines (SVM) [38], decision tree (DT) [39], multi-layer perceptron (MLP) [40] and AdaBoost [29]. For completeness, these classifiers and their hyperparameters are briefly explained. We used the Python library scikit-learn to implement the classifiers in this work.

A. *k*-NEAREST NEIGHBORS (*k*-NN)

The k-NN classifier, proposed in [37], assigns the most frequent pattern among the k nearest training samples to a sample. For instance, if $k = 1$, the assigned pattern is the closest neighbor's. This assignment can be carried out uniformly or weighted according to a distance metric between the samples. In this work, the distance metrics investigated are the Euclidean and Manhattan distance metrics. The latter can be interpreted as the sum of the absolute differences between the samples' Cartesian coordinates.

This method yields a decision surface that can easily adapt to the training data's distribution shape, which is a significant advantage. On the other hand, the main disadvantage is the training phase complexity since many distance metrics should be derived and several neighbors' samples sorted and verified.

B. SUPPORT VECTOR MACHINE (SVM)

SVM is a supervised method that aims to find a hyperplane, also known as a hard margin, that separates the data patterns. Figure 7 shows two patterns (target $= 0$ and target $= 1$) by a hyperplane, $\mathbf{w}^T \cdot \mathbf{x}_i + b = 0$. The points closest to the hyperplane are called support vectors, and $\mathbf{w}$ and b are the coefficients found in the training phase. The objective is to maximize the distances between the hyperplane and the support vectors [41]. The sum of these distances is illustrated as the total margin in Figure 7.

There are some scenarios where the datasets are not separable by a hyperplane. In this case, an adimensional transformation for the input sets using Kernel functions can modify these datasets to be linearly separable. Several Kernel functions are proposed in the literature. This work investigates the linear and the RBF Kernel functions [42]. The latter requires the selection of a hyperparameter, denoted as γ, to control the flexibility of this transformation.

Moreover, we can allow some margin violations by imposing a penalty factor on the optimization problem. This factor, denoted as box constraint C, is a hyperparameter that must be selected. This approach is known as soft-margin SVM. Figure 7 illustrates the soft margins for our case study.

C. DECISION TREE (DT)

A DT, illustrated in Figure 4, is a series of if-else statements created in the training stage that can be used for classification [39]. As depicted in Figure 4, the splitting procedure

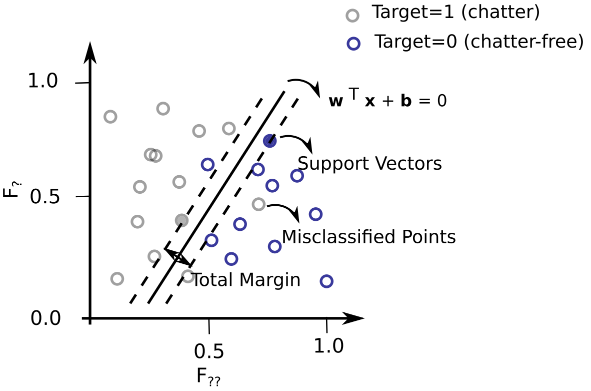


FIGURE 7. Illustration of a soft-margins SVM-based classifier.

starts at the root node, goes through the intermediate nodes, and ends up in the leaves. Moreover, the illustrated tree has three levels.

It is known that the number of samples required to populate the tree doubles for each additional tree level. Therefore, the user can limit the number of levels to prevent overfitting using the hyperparameter max_depth. The user can also decide the number of features that will be considered in each slit decision. This hyperparameter, denoted as max_feature, can also prevent data overfitting. This hyperparameter can be automatically chosen by the procedure or related to the number of features as its square root or logarithm values. For instance, even if 45 features are investigated, only 10 randomly selected features are considered in the splitting procedures if max_feature is set to 10.

D. MULTI LAYER PERCEPTRON (MLP)

An MLP classifier is a feedforward artificial neural network model that maps the input dataset, $\mathbf{x} \in \mathbb{R}^D$, into appropriate targets (outputs), $\mathbf{o} \in \mathbb{R}^L$ [40]. In this work, we investigate the performance of this classifier considering D features and two possible outputs (target $= 0$ or target $= 1$) yield $L = 1$. An MLP is composed of input, hidden, and output layers. The number of hidden layers is a hyperparameter that can vary according to the user's decision.

For instance, a one-hidden-layer MLP can be mathematically described by

$$f(\mathbf{x}) = G\left\{\mathbf{b}^{(2)} + \mathbf{W}^{(2)}\left[s\left(\mathbf{b}^{(1)} + \mathbf{W}^{(1)}\mathbf{x}\right)\right]\right\} \tag{1}$$

with bias vectors $\mathbf{b}^{(1)}$, $\mathbf{b}^{(2)}$; weight matrices $\mathbf{W}^{(1)}$, $\mathbf{W}^{(2)}$ and activation functions G and s.

The vector $h(\mathbf{x}) = s(\mathbf{b}^{(1)} + \mathbf{W}^{(1)}\mathbf{x})$ consists in the hidden layer, while $f(\mathbf{x}) = G[\mathbf{b}^{(2)} + \mathbf{W}^{(2)}h(\mathbf{x})]$. Typical choices for the activation functions are the identity (which returns **a**), the logistic sigmoid function (which $1/(1 + exp(-a))$, the hyperbolic tan function (which returns tanh(a)) and the rectified linear unit function (which returns max(0, a)), considering a the input for the activation functions. In this work, these activation functions are denoted as identity, logistic, tanh, and relu, respectively.

The weight matrices are found by solving an optimization problem during training. Several optimization solvers, including the Limited-Memory BFGS, a quasi-Newton method, or stochastic methods such as stochastic gradient descent and Adam [43], can be used. In this work, these optimization solvers are denoted as lbfgs, sgd, and adam, respectively.

Finally, in this work, we also investigate the use of a regularization term. This term, denoted by alpha, attempts to reduce overfitting by constraining the size of the weights. Increasing alpha may yield simpler decision boundary curves, and decreasing alpha may result in a more complicated decision boundary. More information about the MLP method can be found in [44].

E. AdaBoost

The AdaBoost algorithm works well for classification problems with two classes (targets) [29], but can also be employed in multiclass problems [45]. AdaBoost uses a sequence of weak learners to derive a strong learner. This sequence is derived from several evaluations denoted as boost iterations. A weak learner can typically be formulated as a decision stump, a simple tree composed of a root node and leaves (two leaves for binary classification). In each evaluation, the AdaBoost creates a decision stump using a decision metric, *e.g.* the Gini impurity measure. During the training phase, the next decision stump is created upon weighing the features according to their importance in deriving a good classifier. For example, the probability of a feature being in the following evaluation can be a function of its weight. The algorithm performs many boosting iterations, yielding several decision stumps. The outcome of this classifier is defined according to the most common output from these decision stumps.

The maximum number of evaluations denoted as n_estimators, is a hyperparameter that the user can select. Upon convergence, the algorithm can stop the learning procedure earlier. Moreover, a weight is applied to each decision stump at each boost iteration according to a decision metric. A hyperparameter, denoted as learning_rate, can modify the contribution of each decision stump. There is a trade-off between the learning_rate and n_estimators parameters since a high learning_rate can yield a premature convergence.

F. 1D-CNN

Convolutional Neural Networks (CNNs) are commonly investigated within deep learning, with applications in biomedical signal processing, pattern recognition, autonomous systems, and time-series classification. One-dimensional CNNs (1D-CNNs) have shown utility in modeling sequential data due to their ability to capture localized patterns. These networks have been applied as an alternative to traditional machine learning approaches for signal classification tasks [46].

A 1D-CNN architecture typically consists of two main components: convolutional layers, which extract localized features using trainable filters, and dense (fully connected) layers, which perform classification based on the extracted representations. Relevant hyperparameters include the number and size of filters, pooling strategy, and activation functions. This structure has been used in processing one-dimensional data such as acoustic emission signals, where localized variations can be informative for classification [47].

V. METHODOLOGY

This work investigates micro-milling operations by considering two different materials (AR and UF). The number of samples of each dataset, composed of AE signals during chatter and chatter-free cuts, is described in Table 2. These samples were submitted to the sliding window algorithm, considering 1000 samples as the window size (3ms) and 500 as the step size (1.5ms). This procedure yields n windows.

The features described in Table 1 were extracted and normalized from these windows. These values populate the input vectors for the classifiers, which are $\mathbf{F}_j \in \mathbb{R}^n$, where $j = 1 \ldots 45$. It is essential to highlight that only statistical features have been explored, guaranteeing the generalization of the presented methodology.

Therefore, we trained and tested several classifiers with different hyperparameters at each feature selection phase, as illustrated in Figure 8. The dataset was divided into training and testing using the stratified cross-validation method with k-fold = 5.

The entire set of 45 features is initially used to train and test the classifiers. Then, the RF algorithm identifies N relevant features and the Pearson correlation analysis M features from this set. Finally, collinearity analysis selects a single feature as the most pertinent. The values of N and M can differ for the different materials.

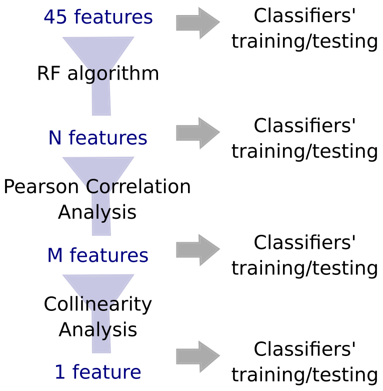

45 features
Classifiers' training/testing
RF algorithm
N features
Classifiers' training/testing
Pearson Correlation Analysis
M features
Classifiers' training/testing
Collinearity Analysis
1 feature
Classifiers' training/testing

FIGURE 8. Illustration of the feature sections phases.

Table 3 describes the investigated classifiers and their hyperparameters. The performance of these classifiers is compared using accuracy, F1-score, training, testing times, and Floating-point Operations per Second (FLOP). This last indicator is directly related to the computational cost of inference and is essential for the analysis of a real implementation on a microcontroller. For this work, the STM32F446RE was considered. The computational routine was developed in Python 3 using the sklearn library, among others. A desktop computer running Windows 10, with Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz and 8GB RAM, carried out the computational analyses.

TABLE 2. Number of data samples for chatter and chatter-free cuts for the different materials.

	Number of points	
Material	Chatter	Non-Chatter
AR	2254322	1502101
UF	1878602	1850139

TABLE 3. Parameters of the classifiers tested in the grid search.

Classifier	Parameters	Range
k-NN	k weights metric	1,3,5,7,13 uniform and distance Euclidean and Manhattan
SVL	C	0.01,0.02,0.05,0.1,0.2,0.25,0.3 0.33,0.4,0.5,0.75,0.9,1,2,3
SVM-RBF	C	0.01,0.02,0.05,0.1,0.2,0.25, 0.3,0.33,0.4,0.5,0.75,0.9,1 2, 3, 5, 10, 20, 50, 100, 150
	γ	0.01,0.02,0.05,0.1,0.2,0.25,0.3 0.33, 0.4, 0.5, 0.75, 0.9, 1, 2, 3
DT	max_depth	1,2,3,4,5,6,7,8,9,10,11 12,13,14,15,16,17,18,19,20
	max_features	auto, sqrt, log2
MLP	alpha solver hidden layers neurons activation	0.0001,0.001,0.01,0.1,1,10 lbfgs, sgd, adam 2 100 identity, logistic, tanh, relu
AdaBoost	n_estimators learning rate	30,50,70,80,90,100,110,120,130 0.0001,0.001,0.01,0.1,1,10

The 1D-CNN model was also trained using the same AE signals and sliding window strategy, with a window size of 1000 samples and a step size of 500 samples. In contrast to traditional classifiers that rely on engineered features, the CNN model operates directly on raw signal segments, learning internal representations from the data. This end-to-end approach reduces the need for manual preprocessing but entails higher computational complexity. The architecture adopted in this study includes two convolutional layers (32 and 64 filters), each followed by max-pooling, and two dense layers, using ReLU activations and a final sigmoid output for binary classification. The model was trained using the Adam optimizer with binary cross-entropy loss and early stopping.

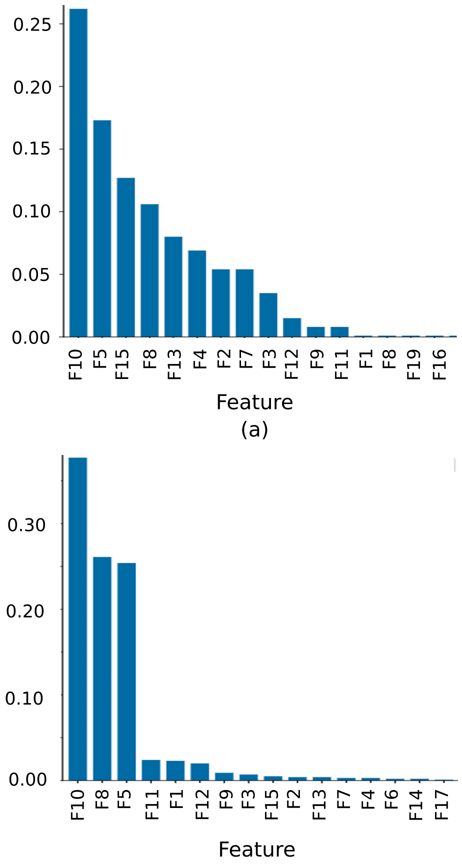
VI. RESULTS AND DISCUSSION

The performance of the classifiers is evaluated and discussed at the end of this section. Firstly, we are presenting the outcome of the feature selection procedures.

From a set of 45 features (described in Table 1), the RF algorithm can derive the most relevant ones for chatter detection according to their relevance. These values are illustrated in bar plots depicted in Figs. 9(a) and (b) for AR and UF materials, respectively. Table 4 summarizes the most relevant features selected by the RF algorithm. This procedure selected $M = 9$ features for the AR and $M = 5$ features for the UF, demonstrating that different materials require the derivation of their classifiers due to the unstable nature of the machining process.

A Pearson correlation analysis can be employed to eliminate redundant information from the selected features of the RF (described in Table 4). The features with a Pearson correlation greater than 0.95 are candidates to be discarded. Figures 10 show the Pearson correlation matrices for both materials. Table 5 reports the pair of candidates for both materials. For example, among the features F_5 and F_{10} for the UF material, F_{10} presents the lower correlation sum, indicating that F_{10} is less redundant than F_3, which will be kept in the further analysis. The AR material presents a tie between F_2 and F_3. In this case, F_2 is selected due to its degree of importance in the RF algorithm (see Figure 9 (a)). Table 4 summarizes the most relevant features the Pearson correlation analysis selected. This procedure selected $N = 4$ features for both materials. Some of the selected features are the same, again demonstrating that different materials may require the derivation of dedicated classifiers.

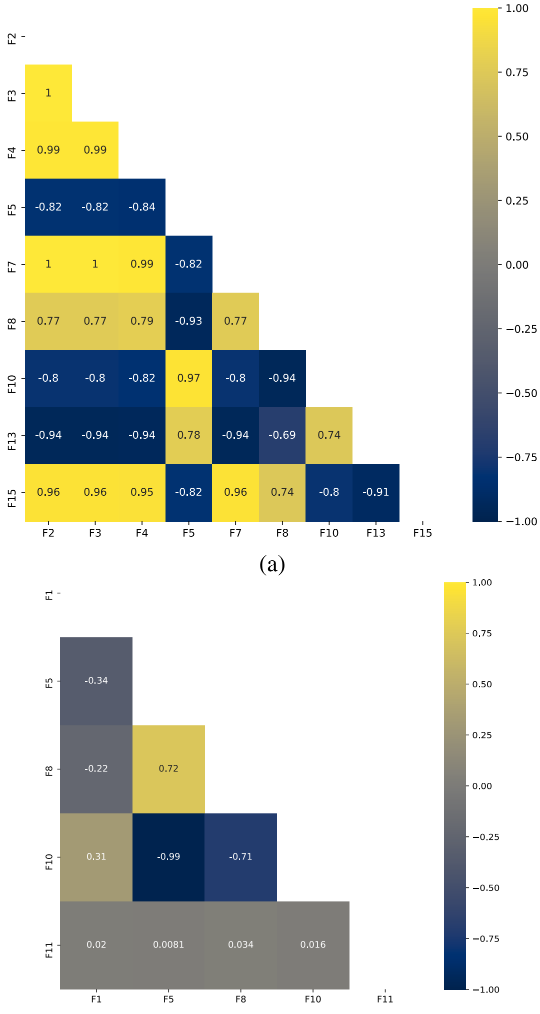
Finally, the collinearity analysis can help demonstrate the separability between the targets (classes), considering $N = 4$ selected features by the correlation analysis $N = 4$. Figures 11 show pairwise evaluations of target values ($target = 1$ for chatter cuts and $target = 0$ for chatter-free cuts), indicating the pair separability regarding the targets.



(a)

(b)

FIGURE 9. **Feature's importance according to RF algorithm: (a) AR and (b) UF.**



(a)

(b)

FIGURE 10. **Pearson correlation matrix of the selected features by RF: (a) AR - $M = 9$, and (b) UF - $M = 5$.**

TABLE 4. **Features selected by Random Forest algorithm, Pearson correlation, and collinearity analyses.**

Algorithm	Mat.	Features
RF	AR	F_2,F_3,F_4,F_5,F_7,F_8,F_{10},F_{13},F_{15}
	UF	F_1, F_5,F_8,F_{10},F_{11}
Pearson	AR	F_8, F_{10}, F_{13}, F_{15}
	UF	F_1, F_8, F_{10}, F_{11}
Collinearity	AR	F_{10}
	UF	F_{10}

TABLE 5. Feature selection by Pearson correlation analysis for both materials.

Mat.	Candidates	$\sum_i^N C_{ij}$		Selected
		1st cand.	2nd cand.	
AR	F_2-F_3	8.28	8.28	F_2
	F_2-F_4	8.28	8.31	F_2
	F_2-F_{15}	8.28	8.1	F_{15}
	F_5-F_{10}	7.80	7.67	F_{10}
	F_7-F_{15}	8.28	8.10	F_{15}
UF	F_5-F_{10}	3.06	3.03	F_{10}

These figures also show the kernel density estimation for each selected feature to illustrate the distribution of the targets.

Figure 11(a) indicates that the target distribution of the features F_8 and F_{10} does not present overlaps. However, the target distributions of the features F_{13} and F_{15} overlap and might not be good candidates for chatter detection during AR machining. Since the standard deviation of the F_{10} distribution is higher than that of the F_8 distribution, F_{10} may represent the data variability in a better manner. Therefore, F_{10} is the feature selected to classify the presence of chatter during AR machining by collinearity analysis as indicated in Table 4.

Figure 11(b) demonstrates that the target distributions of the features F_1, F_8 and F_{11} present overlaps and may not be good candidates for chatter detection during UF machining. On the other hand, since the target distribution of the feature F_{10} does not present overlaps, the feature selected to classify the presence of chatter during UF machining by collinearity analysis is F_{10} as indicated in Table 4.

TABLE 6. Hyperparameters used by classifiers.

Classifier	Parameters	AR	UF
k-NN	neighbors	1	1
	weights	uniform	uniform
	metric	Euclidean	Manhattan
SVM-RBF	C	2	10
	γ	2	0.2
DT	max_depth	5	12
	max_features	sqrt	auto
MLP	alpha	0.0001	0.001
	solver	adam	lbfgs
	activation	logistic	identity
AdaBoost	n_estimators	30	30
	learning rate	0.0001	0.0001

Three feature selection algorithms have been exploited to reduce the dimension of the dataset. The selected features, described in Table 4, are used to train and test five well-known machine learning-based classifiers: k-NN, SVM, DT, MLP, and AdaBoost. Table 3 describes the hyperparameters used during this training. The performance of these classifiers is evaluated using accuracy (ACC), F1-score, training time (T_{train}), and testing time (T_{test}). Table 6 shows the hyperparameters that obtained the best classifiers, while Tables 7 and 8 show the performance index for these classifiers considering the set of features selected in each phase of the dataset dimensionality reduction strategy (see Figure 8).

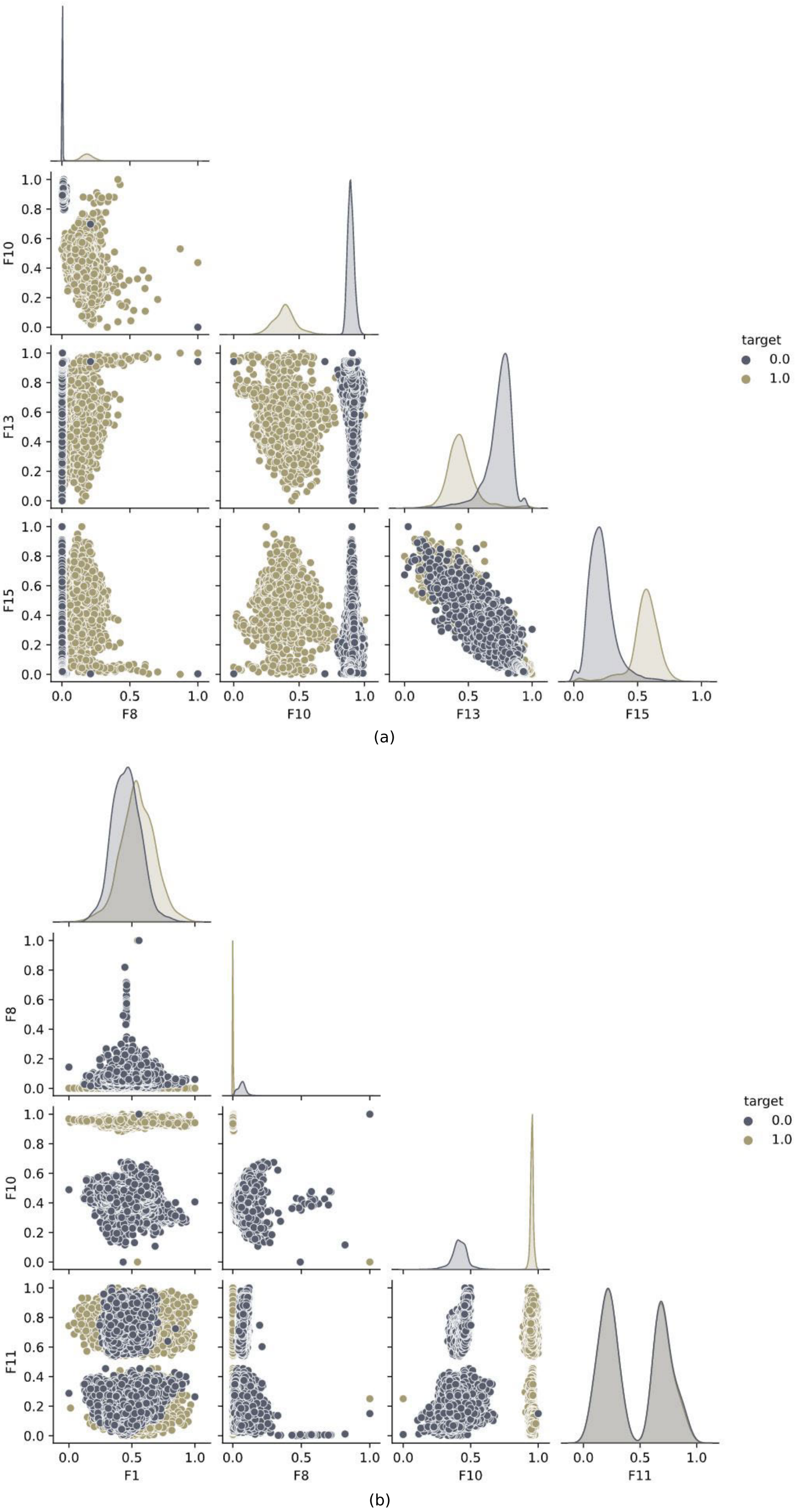
TABLE 7. Classifiers' performance for selected feature sets - AR.

		Performance Indicators			
	Features	ACC	F1Score	T_{train}	T_{test}
k-NN	45	98.67	98.54	–	28825
	9	99.94	99.94	–	277.8
	4	99.95	99.95	–	84.5
	1	100	100	–	79.2
SVM	45	98.33	98.23	23140	12461
	9	99.80	99.78	138.2	96.0
	4	99.92	11.29	82.8	110.7
	1	100	100	100.2	61.2
DT	45	96.74	96.35	14.4	1.6
	9	99.98	99.97	10.8	1.2
	4	99.82	99.81	7.0	1.0
	1	99.98	99.97	3.0	1.0
MLP	45	99.95	99.95	34153	4.8
	9	99.97	99.97	18064	4.2
	4	99.92	99.92	1592.6	2.8
	1	100	100	312.1	3.2
AdaBoost	45	99.99	99.99	11170	18.9
	9	99.99	99.99	341.7	13.4
	4	99.80	99.79	224.0	11.8
	1	100	100	6.2	1.8
1D-CNN	–	99.85	99.84	132.4	6.4

TABLE 8. Classifiers' performance for selected feature sets - UF.

		Performance Indicators			
	Features	ACC	F1-score	T_{train}	T_{test}
k-NN	45	99.65	99.65	–	31646
	5	99.99	99.99	–	97.5
	4	99.98	99.98	–	84.0
	1	100	100	–	80.3
SVM	45	99.86	99.86	42821	16697
	5	100	100	79.9	46.9
	4	99.99	99.99	102.1	65.8
	1	100	100	84.6	55.3
DT	45	88.18	86.06	29.9	3.8
	5	99.99	99.99	4.8	1.0
	4	99.05	99.05	8.2	1.0
	1	99.99	99.99	3.2	1.2
MLP	45	99.98	99.98	26183	5.4
	9	100	100	21208	2.8
	4	99.99	99.99	13675	2.8
	1	100	100	25423	2.4
AdaBoost	45	99.98	99.98	11693	36.2
	5	99.99	99.99	23.9	3.4
	4	99.99	99.99	197.8	10.9
	1	99.99	99.99	13.0	3.8
1D-CNN	–	99.81	99.87	152.0	7.1

According to Tables 7 and 8, the accuracy and F1-score indexes achieve outstanding results considering not only the entire set of features but also the reduced ones. It is important to note that these performance indices present similar values independently of the number of features. This result shows that reducing the number of features added no shortcoming to the classifiers. Moreover, feature selection demonstrates that a single feature (F_{10}) can detect chatter during AR and UF micro-milling operations. Since a single feature can identify chatter, users can adopt a simple decision rule instead of a machine-learning-based classifier.



(a)

(b)

FIGURE 11. **Collinearity analysis with $N = 4$ features: (a) AR, and (b) UF.**

Moreover, by reducing the dataset dimension, the training and testing time presents a critical reduction. This reduction is of utmost importance when implementing strategies for in-process chatter detection. Therefore, DT can be considered the most efficient algorithm for computational effort among classifiers.

Another fundamental analysis for implementing an in-process chatter detection is assessing the required time to extract the features. Table 9 shows the amount of time required for extracting the set of features from the entire dataset ($T_{feature}$) and a window (T_{window} - see Figure 3). A smaller set of features reduces the computational effort necessary, enabling strategies for in-process chatter detection. For instance, our proposal presents window and step sizes of 3ms and 1.5ms, respectively. A DT classifier and a single feature require a shorter time than the window size.

TABLE 9. Analysis of time and FLOPs for both materials.

Material	Features	$T_{feature}$ (s)	T_{window} (ms)	FLOPs ($\times 10^6$)
AR	45	1364.5	83.1	513.2
	9	652.3	39.4	245.3
	5	342.0	0.3	128.7
	4	276.6	6.6	104.1
	1	21.5	1.3	8.1
UF	45	702.4	41.9	265.0
	9	478.8	28.5	180.6
	5	243.4	14.5	91.7
	4	202.4	12.1	76.2
	1	19.7	1.2	7.4

Table 9 also compares the number of floating-point operations per second (FLOPs) – a widely used metric to quantify the computational cost of algorithms – weighted by the volume of data processed at each feature reduction stage. It is observed that the progressive decrease in the dimensionality of the feature set results in a substantial reduction in computational cost, evidenced by the significant drop in FLOPs, without noticeable impairment in the performance indices of the classifiers, such as accuracy and F1-Score. This reduction is particularly relevant in real-time chatter detection contexts during micro-milling operations, where computational resources are limited, as in the case of the STM32F446RE microcontroller. Thus, the FLOP analysis supports the proposal of this paper by demonstrating that a reduced yet informative subset of statistical features significantly contributes to the feasibility of real-time monitoring strategies, meeting the constrained computational requirements without compromising the effectiveness of instability detection in the machining process.

In this study, a 1D-CNN was trained to classify chatter and chatter-free windows directly from raw AE signals, avoiding manual feature extraction. The input to the network consisted of windows of 1000 samples, corresponding to a duration of 3 ms, using the same sliding-window strategy adopted for the traditional classifiers. The adopted 1D-CNN architecture is composed according to Table 10.

The model was trained using the Adam optimizer, with binary cross-entropy loss, a batch size of 128, and early stopping based on validation loss to avoid overfitting.

The total number of trainable parameters in the network was approximately 34,145. A computational cost analysis revealed that the inference phase of the CNN required approximately **1.12 million FLOPs (MFLOPs)** per window. Although this is significantly higher than the computational cost of the traditional classifiers using a reduced feature set (e.g., decision trees or k-NN with a single feature requiring fewer than 10k FLOPs), it remains feasible for deployment in embedded systems with moderate processing capabilities.

In terms of classification performance, the 1D-CNN achieved an F1-Score of 99.84% on the AR dataset and 99.87% on the UF dataset. While these results are comparable to those of the traditional classifiers, they were slightly inferior to some configurations (e.g., DT and k-NN using the F_{10} feature), which reached 100% accuracy and F1-score.

This discrepancy can be attributed to several factors:

- **Data volume:** The amount of training data, while substantial, may still be limited to exploit the learning capacity of the CNN fully.
- **Signal characteristics:** The AE signal exhibits clear amplitude patterns that are well captured by statistical descriptors such as the highest absolute value (F_{10}). These straightforward patterns allow simpler models to separate, reducing the relative advantage of deep feature learning.

Therefore, although 1D-CNN presented high classification capability, its computational cost and marginally lower accuracy suggest that classical machine learning methods -especially with effective feature selection - remain more suitable for real-time deployment for this specific task and dataset.

A. COMPARISON WITH THE RELATED LITERATURE

Reference [48] presented a study addressing in-process chatter detection based on the extraction of micro-milling forces. They apply Variational Mode Decomposition (VMD) to each force group and use the concept that the optimal Intrinsic Mode Function (IMF) is adopted based on the Laplacian Score (LS). As input to the SVM classifier, they use the Multi-scale Permutation Entropy (MSPE) values of the optimal IMF of each group. In summary, the classifier achieved 85% accuracy.

Similarly, [49] emphasize that environmental noise should be considered in chatter detection in micro-milling processes. In this regard, the authors propose a methodology that employs the Variable Forgetting Factor Recursive Least-Squares (VFF-RLS) algorithm to filter out the chatter-independent component, which consists of noise and periodic components. Using the SVM classifier, they achieved an accuracy above 99.1%.

A study considering features obtained in the frequency domain and time domain and extracted using the Stacked-Denoising Autoencoder (SDAE) algorithm for chatter classification in micro-milling processes through the intelligent improved Adaboost-SVM classifier was conducted by [12]. The authors achieved an accuracy of around 96% for the training set.

In [10], the authors proposed extracting nine features and applied the Perceptron classifier and the SVM with an RBF kernel. After selecting two features through a metric

TABLE 10. 1D-CNN architecture adopted with several trainable parameters and estimated FLOPs per layer.

Layer	Parameters	Output shape	Estimated FLOPs
Input	–	(1000, 1)	–
Conv1D #1 (32 filters, kernel=5)	$(5 \times 1 + 1) \times 32 = \mathbf{192}$	(996, 32)	$996 \times 32 \times 5 \times 2 = \mathbf{318,720}$
MaxPooling1D #1 (pool=2)	–	(498, 32)	–
Conv1D #2 (64 filters, kernel=3)	$(3 \times 32 + 1) \times 64 = \mathbf{6,208}$	(496, 64)	$496 \times 64 \times 3 \times 32 \times 2 = \mathbf{6,094,848}$
MaxPooling1D #2 (pool=2)	–	(248, 64)	–
Flatten	–	(15,872)	–
Dense #1 (64 units)	$15,872 \times 64 + 64 = \mathbf{1,015,616}$	(64)	$15,872 \times 64 \times 2 = \mathbf{2,031,616}$
Dense #2 (1 unit, Sigmoid)	$64 \times 1 + 1 = \mathbf{65}$	(1)	$64 \times 1 \times 2 = \mathbf{128}$
Total	**1,022,081**	–	**8,445,312**

involving Pearson's correlation, the Perceptron achieved 54.6% accuracy, and the SVM-RBF achieved 99.1%.

It is observed that this work presented a sequence of already established techniques, and the result indicates that only the feature F_{10} (Highest Absolute Value) is sufficient for chatter classification. Moreover, none of the related works presented metrics for a real implementation on a microcontroller (FLOPs). Thus, it is evident that the contributions of this work are justified.

VII. CONCLUSION

Chatter, characterized as undesired vibrations, promotes the improper finishing of parts, the reduction of the life of cutters and machines, and the waste of energy and materials. However, several studies mitigate the effects of this phenomenon by predicting or detecting it using dynamic modeling and stability lobes. In addition, chatter detection strategies may use machine learning classifiers, leading to tight computational requirements.

A smaller yet relevant set of features could support these classifiers in coping with these requirements. This work proposed using feature selection to evaluate the impact of several statistical features on the performance of machine learning-based classifiers for chatter detection. Using AR and UF materials, we exploited Random Forest, Pearson's correlation, and collinearity analysis to identify the most relevant features in micro-milling operations. In this way, three feature reduction stages have been implemented, reducing the number of features from 45 to one single.

We evaluated five machine learning-based classifiers in each feature reduction stage (k-NN, SVM, DT, MLP, and AdaBoost) and a deep learning classifier. The performance of these classifiers was compared according to accuracy, F1-Score, training, testing time, and FLOPs. The accuracy and F1-Score indices present similar values independently of the number of features. This result shows that reducing the number of features added has no impact on the classifiers. Moreover, by reducing the dataset dimension, the training, testing time, and FLOPs present a critical reduction. This reduction is of utmost importance when implementing strategies for in-process chatter detection.

Incorporating a 1D-CNN architecture in this study contributed to evaluating deep learning methods for chatter classification. However, the results suggest that simpler classifiers combined with appropriate feature selection may offer comparable accuracy with significantly lower computational costs when computational resources are limited. This trade-off is particularly relevant for embedded systems, such as those based on microcontrollers, where inference time and energy efficiency are critical design considerations.

REFERENCES

[1] L. O'Toole, C.-W. Kang, and F.-Z. Fang, "Precision micro-milling process: State of the art," *Adv. Manuf.*, vol. 9, no. 2, pp. 173–205, Oct. 2020, doi: 10.1007/s40436-020-00323-0.

[2] X. Jing, Z. Zheng, J. Xu, F. Wang, S. H. I. Jaffery, and H. Li, "Stability analysis in micro milling based on p-leader multifractal method," *J. Manuf. Processes*, vol. 77, pp. 495–507, May 2022, doi: 10.1016/j.jmapro.2022.03.033.

[3] S. S. Park and R. Rahnama, "Robust chatter stability in micro-milling operations," *CIRP Ann.*, vol. 59, no. 1, pp. 391–394, 2010, doi: 10.1016/j.cirp.2010.03.023.

[4] G. Quintana and J. Ciurana, "Chatter in machining processes: A review," *Int. J. Mach. Tools Manuf.*, vol. 51, no. 5, pp. 363–376, May 2011, doi: 10.1016/j.ijmachtools.2011.01.001.

[5] Y. Fu, Y. Zhang, H. Zhou, D. Li, H. Liu, H. Qiao, and X. Wang, "Timely online chatter detection in end milling process," *Mech. Syst. Signal Process.*, vol. 75, pp. 668–688, Jun. 2016, doi: 10.1016/j.ymssp.2016.01.003.

[6] G. Quintana, J. Ciurana, and D. Teixidor, "A new experimental methodology for identification of stability lobes diagram in milling operations," *Int. J. Mach. Tools Manuf.*, vol. 48, no. 15, pp. 1637–1645, Dec. 2008, doi: 10.1016/j.ijmachtools.2008.07.006.

[7] J. Friedrich, J. Torzewski, and A. Verl, "Online learning of stability lobe diagrams in milling," *Proc. CIRP*, vol. 67, pp. 278–283, Jan. 2018, doi: 10.1016/j.procir.2017.12.213.

[8] F. Finkeldey, A. Saadallah, P. Wiederkehr, and K. Morik, "Real-time prediction of process forces in milling operations using synchronized data fusion of simulation and sensor data," *Eng. Appl. Artif. Intell.*, vol. 94, Sep. 2020, Art. no. 103753, doi: 10.1016/j.engappai.2020.103753.

[9] J. Shi, X. Jin, and H. Cao, "Chatter stability analysis in micro-milling with aerostatic spindle considering speed effect," *Mech. Syst. Signal Process.*, vol. 169, Apr. 2022, Art. no. 108620, doi: 10.1016/j.ymssp.2021.108620.

[10] G. S. Sestito, G. S. Venter, K. S. B. Ribeiro, A. R. Rodrigues, and M. M. da Silva, "In-process chatter detection in micro-milling using acoustic emission via machine learning classifiers," *Int. J. Adv. Manuf. Technol.*, vol. 120, nos. 11–12, pp. 7293–7303, Apr. 2022, doi: 10.1007/s00170-022-09209-w.

[11] L. Zhu and C. Liu, "Recent progress of chatter prediction, detection and suppression in milling," *Mech. Syst. Signal Process.*, vol. 143, Sep. 2020, Art. no. 106840, doi: 10.1016/j.ymssp.2020.106840.

[12] S. Wan, X. Li, Y. Yin, and J. Hong, "Milling chatter detection by multi-feature fusion and AdaBoost-SVM," *Mech. Syst. Signal Process.*, vol. 156, Jul. 2021, Art. no. 107671, doi: 10.1016/j.ymssp.2021.107671.

[13] W.-K. Wang, M. Wan, W.-H. Zhang, and Y. Yang, "Chatter detection methods in the machining processes: A review," *J. Manuf. Processes*, vol. 77, pp. 240–259, May 2022, doi: 10.1016/j.jmapro.2022.03.018.

[14] Y. Fu, Y. Zhang, Y. Gao, H. Gao, T. Mao, H. Zhou, and D. Li, ''Machining vibration states monitoring based on image representation using convolutional neural networks,'' *Eng. Appl. Artif. Intell.*, vol. 65, pp. 240–251, Oct. 2017, doi: 10.1016/j.engappai.2017.07.024.

[15] F. Shi, H. Cao, X. Zhang, and X. Chen, ''A reinforced k-nearest neighbors method with application to chatter identification in high-speed milling,'' *IEEE Trans. Ind. Electron.*, vol. 67, no. 12, pp. 10844–10855, Dec. 2020, doi: 10.1109/TIE.2019.2962465.

[16] K. S. B. Ribeiro, G. S. Venter, and A. R. Rodrigues, ''Experimental correlation between acoustic emission and stability in micromilling of different grain-sized materials,'' *Int. J. Adv. Manuf. Technol.*, vol. 109, nos. 7–8, pp. 2173–2187, Jul. 2020.

[17] A. Sio-Sever, E. Leal-Muñoz, J. Lopez-Navarro, R. Alzugaray-Franz, A. Vizan-Idoipe, and G. de Arcas-Castro, ''Non-invasive estimation of machining parameters during end-milling operations based on acoustic emission,'' *Sensors*, vol. 20, no. 18, p. 5326, Sep. 2020, doi: 10.3390/s20185326.

[18] H. Hegab, U. Umer, I. Deiab, and H. Kishawy, ''Performance evaluation of Ti–6Al–4 V machining using nano-cutting fluids under minimum quantity lubrication,'' *Int. J. Adv. Manuf. Technol.*, vol. 95, nos. 9–12, pp. 4229–4241, Jan. 2018, doi: 10.1007/s00170-017-1527-z.

[19] H. A. Kishawy, H. Hegab, U. Umer, and A. Mohany, ''Application of acoustic emissions in machining processes: Analysis and critical review,'' *Int. J. Adv. Manuf. Technol.*, vol. 98, nos. 5–8, pp. 1391–1407, Jun. 2018, doi: 10.1007/s00170-018-2341-y.

[20] B. Sener, M. U. Gudelek, A. M. Ozbayoglu, and H. O. Unver, ''A novel chatter detection method for milling using deep convolution neural networks,'' *Measurement*, vol. 182, Sep. 2021, Art. no. 109689, doi: 10.1016/j.measurement.2021.109689.

[21] M. C. Yesilli, F. A. Khasawneh, and A. Otto, ''Chatter detection in turning using machine learning and similarity measures of time series via dynamic time warping,'' *J. Manuf. Processes*, vol. 77, pp. 190–206, May 2022, doi: 10.1016/j.jmapro.2022.03.009.

[22] P.-H. Kuo, M.-J. Huang, P.-C. Luan, and H.-T. Yau, ''Study on bandwidth analyzed adaptive boosting machine tool chatter diagnosis system,'' *IEEE Sensors J.*, vol. 22, no. 9, pp. 8449–8459, May 2022, doi: 10.1109/JSEN.2022.3163914.

[23] M.-Q. Tran, M. Elsisi, and M.-K. Liu, ''Effective feature selection with fuzzy entropy and similarity classifier for chatter vibration diagnosis,'' *Measurement*, vol. 184, Nov. 2021, Art. no. 109962, doi: 10.1016/j.measurement.2021.109962.

[24] J. Ye, P. Feng, C. Xu, Y. Ma, and S. Huang, ''A novel approach for chatter online monitoring using coefficient of variation in machining process,'' *Int. J. Adv. Manuf. Technol.*, vol. 96, nos. 1–4, pp. 287–297, Jan. 2018, doi: 10.1007/s00170-017-1544-y.

[25] M. Lamraoui, M. Barakat, M. Thomas, and M. E. Badaoui, ''Chatter detection in milling machines by neural network classification and feature selection,'' *J. Vibrat. Control*, vol. 21, no. 7, pp. 1251–1266, Aug. 2013, doi: 10.1177/1077546313493919.

[26] X. Wang, C. Fidge, G. Nourbakhsh, E. Foo, Z. Jadidi, and C. Li, ''Anomaly detection for insider attacks from untrusted intelligent electronic devices in substation automation systems,'' *IEEE Access*, vol. 10, pp. 6629–6649, 2022, doi: 10.1109/ACCESS.2022.3142022.

[27] R. Rafal, L. Pawel, K. Krzysztof, K. Bogdan, and W. Jerzy, ''Chatter identification methods on the basis of time series measured during titanium superalloy milling,'' *Int. J. Mech. Sci.*, vol. 99, pp. 196–207, Aug. 2015, doi: 10.1016/j.ijmecsci.2015.05.013.

[28] U. M. Khaire and R. Dhanalakshmi, ''Stability of feature selection algorithm: A review,'' *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 4, pp. 1060–1073, Apr. 2022, doi: 10.1016/j.jksuci.2019.06.012.

[29] L. Breiman, ''Random forests,'' *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/a:1010933404324.

[30] R. Genuer, J.-M. Poggi, and C. Tuleau-Malot, ''Variable selection using random forests,'' *Pattern Recognit. Lett.*, vol. 31, no. 14, pp. 2225–2236, Oct. 2010, doi: 10.1016/j.patrec.2010.03.014.

[31] X. Li, W. Chen, Q. Zhang, and L. Wu, ''Building auto-encoder intrusion detection system based on random forest feature selection,'' *Comput. Secur.*, vol. 95, Aug. 2020, Art. no. 101851, doi: 10.1016/j.cose.2020.101851.

[32] J.-X. Liang, J.-F. Zhao, N. Sun, and B.-J. Shi, ''Random forest feature selection and back propagation neural network to detect fire using video,'' *J. Sensors*, vol. 2022, pp. 1–10, Jan. 2022, doi: 10.1155/2022/5160050.

[33] W. A. Souza, A. M. S. Alonso, T. B. Bosco, F. D. Garcia, F. A. S. Gonçalves, and F. P. Marafão, ''Selection of features from power theories to compose NILM datasets,'' *Adv. Eng. Informat.*, vol. 52, Apr. 2022, Art. no. 101556, doi: 10.1016/j.aei.2022.101556.

[34] K. Rawal and A. Ahmad, ''Feature selection for electrical demand forecasting and analysis of Pearson coefficient,'' in *Proc. IEEE 4th Int. Electr. Energy Conf. (CIEEC)*, May 2021, pp. 1–6, doi: 10.1109/CIEEC50170.2021.9510614.

[35] G. S. Sestito, A. C. Turcato, A. L. Dias, P. Ferrari, D. H. Spatti, and M. M. da Silva, ''A general optimization-based approach to the detection of real-time Ethernet traffic events,'' *Comput. Ind.*, vol. 128, Jun. 2021, Art. no. 103413. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361521000208

[36] X. Tang, Y. Dai, Q. Liu, X. Dang, and J. Xu, ''Application of bidirectional recurrent neural network combined with deep belief network in short-term load forecasting,'' *IEEE Access*, vol. 7, pp. 160660–160670, 2019, doi: 10.1109/ACCESS.2019.2950957.

[37] T. M. Cover and P. E. Hart, ''Nearest neighbor pattern classification,'' *IEEE Trans. Inf. Theory*, vol. IT-13, no. 1, pp. 21–27, Jan. 1967.

[38] V. Vapnik, *The Support Vector Method of Function Estimation*. Boston, MA, USA: Springer, 1998, ch. 3, pp. 55–85.

[39] S. R. Safavian and D. Landgrebe, ''A survey of decision tree classifier methodology,'' *IEEE Trans. Syst. Man, Cybern.*, vol. 21, no. 3, pp. 660–674, Jun. 1991.

[40] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, ''Learning internal representations by error propagation,'' in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*, 1987, pp. 318–362.

[41] L. Wang, Ed., *Support Vector Machines* (Studies in fuzziness and soft computing). Berlin, Germany: Springer, Jan. 2005.

[42] I. Steinwart and A. Christmann, *Support Vector Machines* (Information Science and Statistics). New York, NY, USA: Springer, Aug. 2008.

[43] D. P. Kingma and J. Ba, ''Adam: A method for stochastic optimization,'' 2014, *arXiv:1412.6980*.

[44] T. Windeatt, ''Accuracy/diversity and ensemble MLP classifier design,'' *IEEE Trans. Neural Netw.*, vol. 17, no. 5, pp. 1194–1211, Sep. 2006.

[45] T. Hastie, S. Rosset, J. Zhu, and H. Zou, ''Multi-class AdaBoost,'' *Statist. Interface*, vol. 2, no. 3, pp. 349–360, 2009.

[46] A. O. Ige and M. Sibiya, ''State-of-the-art in 1D convolutional neural networks: A survey,'' *IEEE Access*, vol. 12, pp. 144082–144105, 2024.

[47] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, ''1D convolutional neural networks and applications: A survey,'' *Mech. Syst. Signal Process.*, vol. 151, Apr. 2021, Art. no. 107398, doi: 10.1016/j.ymssp.2020.107398.

[48] P. Wang, Q. Bai, K. Cheng, Y. Zhang, L. Zhao, and H. Ding, ''Investigation on an in-process chatter detection strategy for micro-milling titanium alloy thin-walled parts and its implementation perspectives,'' *Mech. Syst. Signal Process.*, vol. 183, Jan. 2023, Art. no. 109617, doi: 10.1016/j.ymssp.2022.109617.

[49] M. Wan, W.-K. Wang, W.-H. Zhang, and Y. Yang, ''Chatter detection for micro milling considering environment noises without the requirement of dominant frequency,'' *Mech. Syst. Signal Process.*, vol. 199, Sep. 2023, Art. no. 110451, doi: 10.1016/j.ymssp.2023.110451.

GUILHERME SERPA SESTITO received the degree in electrical engineering and the master's and Ph.D. degrees from the University of São Paulo (USP), in 2011, 2015, and 2018, respectively. His master's thesis and doctoral dissertation focused on machine learning applied to industrial problems, particularly in communication networks. He worked for approximately eight years in the consulting and product development sector, specializing in industrial communication protocols. Currently, he is a Professor with the Department of Electrical Engineering, Federal University of Technology—Paraná (UTFPR), Cornélio Procópio, Brazil. He has participated in the program of the Department of Mechanical Engineering, USP, through a postdoctoral fellowship funded by CAPES (PNPD).

WESLEY ANGELINO DE SOUZA (Member, IEEE) was born in Brazil, in 1985. He received the Ph.D. degree in electrical engineering from the School of Electrical and Computer Engineering, State University of Campinas (FEEC/UNICAMP), Campinas, SP, Brazil, in 2016. He was a Guest Researcher with the University of Padova (UNIPd/Italy), Padua, Italy, from 2012 to 2013. He a Postdoctoral Researcher with FEEC/UNICAMP; Computer Department, the Federal University of São Carlos (DComp/UFSCar), Sorocaba, SP; and the Institute of Science and Technology—Sorocaba, State University of São Paulo (ICTS/UNESP), Sorocaba, in 2017, 2019, and 2020, respectively. Since 2020, he has been an Associate Professor with the Electrical Engineering Department, Federal University of Technology—Paraná (UTFPR), Cornélio Procópio, Brazil. His expertise includes embedded systems, power meters, smart meters, machine learning, and low/high-level programming.

ALESSANDRO ROGER RODRIGUES received the bachelor's and master's degrees in mechanical engineering from São Paulo State University (UNESP), in 1998 and 2001, respectively, and the Ph.D. degree in mechanical engineering from the University of São Paulo (USP), in 2005. He completed a PDEE/CAPES sandwich doctoral internship with the Technische Universität Darmstadt (TUD), Germany, in 2004. He is currently an Associate Professor with the São Carlos School of Engineering (EESC), USP. He teaches undergraduate and graduate courses in the field of mechanical manufacturing. His main research areas include surface integrity and machinability in macro and micro mechanical machining of metallic materials.

MAÍRA MARTINS DA SILVA received the bachelor's and master's degrees in mechanical engineering from the University of São Paulo (EESC-USP), in 2001 and 2004, respectively, and the Ph.D. degree in mechanical engineering from Katholieke Universiteit Leuven, Belgium, in 2009. She is an Associate Professor with the Department of Mechanical Engineering, São Carlos School of Engineering (EESC-USP). In 2010, she was a Postdoctoral Researcher with the Department of Mechanical Production Engineering, EESC-USP, with funding from FAPESP. In 2018, she received the Livre-Docência (a postdoctoral academic qualification in Brazil). Her Ph.D. study was supported by CAPES. Her experience lies in the fields of mechanical and mechatronic engineering, focusing mainly on mechatronic product development, multiphysics simulation, control of robotic systems, optimization, and machine learning techniques.

Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) - ROR identifier: 00x0ma614