IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 December 2025, accepted 3 January 2026, date of publication 12 January 2026, date of current version 15 January 2026.

Digital Object Identifier 10.1109/ACCESS.2026.3652205

RESEARCH ARTICLE

Milling Tool Wear Estimation Based on a Neuro Fuzzy Algorithm and Micro Evolutionary Particle Swarm Optimization

GIOVANNI O. DE SOUSA[1], PAULO M. C. MONSON[1], FÁBIO R. L. DOTTO[1], DENNIS BRANDAO[2], (Member, IEEE), AND PEDRO O. C. JUNIOR[1], (Senior Member, IEEE)

[1]Department of Electrical and Computer Engineering, University of São Paulo (EESC-USP), São Carlos 13566-590, Brazil

[2]Department of Information Engineering, University of Brescia, 25123 Brescia, Italy

Corresponding author: Pedro O. C. Junior (pedro.oliveiracjr@usp.br)

This work was supported in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior–Brasil (CAPES) under Grant 001, in part by the National Council for Scientific and Technological Development (CNPq) under Grant 445054/2024-7, in pat by São Paulo Research Foundation (FAPESP) under Grant 2023/02413-2, and in part by the Pro-Rectory of Research and Innovation of the University of São Paulo under Grant 22.1.09345.01.2.

ABSTRACT Milling is one, if not the most important, manufacturing process based on machining in today's industry. Since this is a continuous process, mechanical tool wear becomes an active concern that affects the final product quality. Tool condition monitoring (TCM) is critical in milling to enhance product quality and prevent tool failures, while tool wear estimation is the first step to achieving a complete system. Several intelligent models have been proposed in the literature to estimate mechanical tool wear, but the use of different combinations of recent algorithms in a hybrid way has not been entirely explored. This study introduces a hybrid tool wear estimation model for milling, combining an Adaptive Neuro-Fuzzy Inference System (ANFIS) with Micro Evolutionary Particle Swarm Optimization (MEPSO) on both antecedent and consequent parameters. MEPSO is a recent modification of the classic particle swarm optimization that has not yet been applied in tool wear estimation or in a hybrid setting with ANFIS. The proposed hybrid ANFIS-MEPSO model addresses tool wear prediction challenges by optimizing ANFIS parameters through MEPSO while employing a feature selection method in the ANFIS-MEPSO inputs. A dataset containing experimental milling data, including vibration, acoustic emission, and electrical current signals, was used to train and test the model with 8 features extracted and a feature selection algorithm to estimate tool wear, achieving promising results with MSE, MAE, and R^2 metrics of 0.0045, 0.0578, and 0.9058 on testing data. A sensitivity analysis was conducted to better understand the changes in performance by modifying the optimization hyperparameters. It was revealed that stable and optimal performance occurred with σ and γ between 0.6–0.8 and at least 20 particles, while higher values (0.9) led to inconsistent results. This research highlights ANFIS-MEPSO's viability for tool wear monitoring in milling, suggesting potential for broader application and further refinement.

INDEX TERMS Micro evolutionary particle swarm optimization, milling, neuro-fuzzy, tool wear estimation.

I. INTRODUCTION

The milling process stands out as one of the most significant machining processes in the industry, given its widespread use and the continuous drive to improve efficiency, safety, and product quality while minimizing costs. Tool condition monitoring (TCM) systems are essential to prevent tool failures and ensure the quality of the final product [1]. Several TCM systems were proposed, ranging from direct methods such as vision systems or microscopes to indirect methods

The associate editor coordinating the review of this manuscript and approving it for publication was S. K. Raghuwanshi.

that measure and process correlated machining parameters and signals for an effective understanding of real-time tool wear [2].

Several works were focused on the latter, specifically trying to establish new artificial intelligence (AI) models to estimate tool wear or condition based on signals and sensors measuring acoustic emission (AE), vibration, or cutting force, to name a few. Tool wear estimation is often driven by AI, particularly supervised machine learning algorithms. Classic algorithms, such as decision trees and k-nearest neighbors, are used for classification and linear regression. More advanced learning models, including convolutional and recurrent neural networks, are also employed. Additionally, human-inspired models, such as shallow neural networks and fuzzy logic systems, have shown promising results. Among these, the adaptive neuro-fuzzy inference system (ANFIS) stands out for combining the flexibility and interpretability of fuzzy logic with neural network-like parameter training techniques [3].

Numerous studies have explored the use of AI models to predict tool wear during milling operations, which may or may not include experiments to aid the validation of the algorithm. Experiments were proposed by [4] in aluminum matrix composite milling, using a multilayer perceptron to estimate tool flank wear and tool corner wear. Their results were mixed, with some metrics achieving less than expected performance. Artificial neural networks (ANN), support vector machines (SVM), and Gaussian process regression (GPR) were proposed by [5] to estimate tool wear with multiple sensor data and feature extraction in the time and frequency domains, achieving interesting results in some algorithms while utilizing more classical machine learning models. All data came from two different datasets of milling experiments fused together.

Advanced machine learning algorithms based on deep learning were also tested in a milling tool wear estimation setting. Convolutional neural networks (CNN) were applied in synthetic feature extraction and estimation of remaining useful life (RUL) based on tool wear in a milling dataset [6]. Their main contribution was a novel holistic approach to use said CNNs in a more effective way. Multidimensional stacked sparse autoencoders (SSAE) and a modified loss function were employed by [7] to predict tool wear by combining different feature domains as inputs on vibration signals. Their model was validated by real manufacturing milling experiments, achieving results that surpassed regular SSAE and other machine learning algorithms. A similar model was proposed by [8] with SSAE and a backpropagation neural network to estimate tool wear, using multi-sensor feature extraction and selection based on the Pearson correlation coefficient (PCC). Other classical models were compared, and the proposed method was more effective and produced the best results. A recent yet distinct approach employed a SSAE combined with a gradient boosting decision tree (GBDT) regression as a secondary learner, with Bayesian hyperparameter optimization, for milling tool-wear prediction [9]. Multi-sensor data features were selected using the PCC, and the model's performance was compared against baseline methods, demonstrating enhanced prediction accuracy and reliability.

Physics-informed machine learning and physics-based models were also used in tool wear estimation to enhance their predictive capabilities. A deep learning physics-based approach to tool wear prediction took advantage of tool wear descriptions based on the complex physics involved, improving the deep learning model's capabilities in several areas [10]. Based on experimental validation, their physics-informed results surpassed several machine learning and deep learning models. A physics-guided deep learning model combined a gated recurrent unit (GRU) network with physical constraints on its loss function in a high-speed milling application to estimate tool wear, thereby improving performance and reducing physical inconsistency in the prediction model [11]. The model was validated by experimental and benchmark tests, and the proposed model was compared to other machine learning models and a physical milling model, resulting in good performance and better physical consistency than non physical-assisted prediction models. A gray-box modeling approach applied in high-speed milling tool wear prediction. This approach combines a glass box finite element simulation to extract physically relevant features, milling process parameters, and features extracted from cutting force sensor signals that were conditionally used as inputs in closed box machine learning models [12]. The feature selection method based on the analysis of variance (ANOVA) and the whale optimization algorithm (WOA) had a positive effect on performance in the support vector regression (SVR) model and the gradient boosting regressor (GBR). The GBR model metrics were lower without the use of glass box physical features, highlighting their impact on the approach.

Many hybrid algorithms, i.e., combinations of two or more different models, were explored to estimate tool wear. Usually, in these combinations, a base model performs the estimation while another algorithm optimizes its parameters or hyperparameters. An extreme learning machine (ELM) model optimized with genetic algorithms (GA), particle swarm optimization (PSO), and a combination of both was introduced to estimate tool wear, with time, frequency, and time-frequency domain extracted features as inputs [13]. The proposed GAPSO approach outperformed the ELM with classical GA or PSO optimization. A SVM model with gray wolf optimization (GWO) was applied by [14] to estimate tool condition while combining GA with PCC for feature selection. The GWO improved the method's efficiency, while the GA-based feature selection benefited all models. Another SVM model with a WOA was proposed by [15] to estimate milling tool wear using multi-dimensional features from multi-sensor data as inputs, along with PCC-based feature selection, resulting in a superior combination when compared to GA or PSO optimization. A SVR model was also used to estimate tool wear, with a combination of GWO and differential evolution (DE) algorithms to optimize its parameters. Features from time, frequency, and time-frequency domains were extracted from vibration and cutting power signals, and a selection based on PCC and F-score served as inputs [16]. Experimental data validated the model, and comparisons with deep learning algorithms showed substantially faster test times with similar accuracy. A bidirectional long short-term memory (BiLSTM) network and a GA optimization of its hyperparameters were utilized to predict tool wear in ball nose tungsten carbide cutters during milling [17]. Their research focused on the performance gain from the use of the long short-term memory (LSTM) network to a GA optimized BiLSTM, resulting in a 40% to 60% decrease in error metrics in tool wear prediction. A novel enhanced african vulture optimization algorithm (TDLAVOA) was integrated with extreme gradient boosting (XGBoost) to optimize seven critical hyperparameters for tool wear prediction in milling and turning operations [18]. The model was validated through milling simulations, achieving superior performance compared with previous similar approaches. Finally, [19] proposed an ANFIS model with modified PSO for optimizing cutting parameters and predicting tool wear in milling, using both process parameters and sensor features as inputs without feature selection, yielding superior results compared to classical methods.

Modifying an existing algorithm is also not unheard of. A hybrid version of PSO was proposed by [20], namely micro evolutionary PSO (MEPSO), based on the concept of micro population and evolutionary operators. This optimization technique was recently introduced in the literature and has not yet been fully explored. MEPSO has still not been applied in tool wear estimation for milling, nor in a hybrid combination with ANFIS. However, similar research supports the applicability of this new model in tool wear estimation, with ANFIS serving as the base prediction model and MEPSO optimizing its parameters.

To the best of authors' knowledge, this work presents the first application of a hybrid model combining ANFIS and MEPSO applied to tool wear estimation in milling. Its main objective is to assess the viability of combining such algorithms in a machine learning prediction using real milling experimental data. A similar work by [19] used ANFIS with a hybrid version of PSO based on vibration to estimate tool wear in milling, employing only cutting force and parameters as inputs. The main differences here are the use of a different PSO variant, namely MEPSO, and the application of a multi-sensor approach with feature selection based on PCC to choose the most correlated inputs. Therefore, the main contributions of this paper are:

- The experimentation of a hybrid algorithm (ANFIS-MEPSO) for accurate tool wear estimation, coupled with multi-sensor data and PCC-based feature selection, ensuring that only physically meaningful inputs are used;
- A comprehensive comparative analysis of the model's performance with recent hybrid algorithms and baseline models to quantify the improvements of MEPSO optimization;
- A sensitivity analysis of the MEPSO hyperparameters, including population size, mutation magnitude (σ), and crossover amplitude (γ), to evaluate the model's accuracy and convergence behavior, providing an understanding of the algorithm's internal dynamics, which is rarely explored in similar studies and strengthens the reproducibility and interpretability of the proposed hybrid model.

All these contributions are framed within an intelligent industrial computing application focused on improving manufacturing processes.

The presented work used an experimental dataset. Time-domain features were extracted from all sensor signals, namely AE, vibration, and electrical current. The most correlated features were identified using the PCC and were selected as the inputs for training and testing the ANFIS-MEPSO model. The mean squared error (MSE), mean absolute error (MAE), and R^2 metrics were used to evaluate the model's performance. This combination represents an unified framework integration of signal processing, feature engineering, and evolutionary optimization within a unified framework for intelligent manufacturing systems.

II. META HEURISTIC OPTIMIZATION

Optimization techniques are important to several engineering fields and technologies, including machining and TCM systems. Traditional optimization consists of finding the optimal solution in a mathematically well-defined linear problem that includes a differentiable objective function. However, real-world problems are neither simple, linear, nor well-defined. Therefore, meta-heuristic (MH) methods are useful for finding approximate solutions that may be suitable for specific complex problems. MH algorithms use stochastic operators and can be classified based on the inspiration behind their conception, such as evolutionary, physics-based, swarm-based, or human-based approaches [21]. Various studies in tool wear estimation research have focused on evolutionary and swarm-based MHs to optimize algorithms or other parameters, achieving efficient models.

Genetic algorithms (GA) were conceived based on the evolutionary idea that the most adaptable organisms survive and pass on their genes to the next generation. This concept was translated into an optimization technique, implementing biological concepts such as mutation and crossover in a computational algorithmic manner. A GA starts by generating a random initial population of individuals called chromosomes. Each chromosome has a fitness value that reflects its performance in solving an individual problem based on an objective function (OF). The next generation retains a percentage of the most fit individuals from the previous generation based on a selection operator, and new chromosomes are created via crossover (merging two chromosomes) and mutation (random modifications) [22].

Particle Swarm Optimization (PSO) is an MH inspired by the paradigm of mutual collaboration among biological individuals in swarms to achieve specific goals, such as birds or insects flocking together. This is formalized as swarm intelligence. PSO is composed of particles that have two main components: position and velocity. The position indicates the particle's location in the solution space for a specific optimization problem, while the velocity updates its position using a stochastic equation. Each iteration begins with a randomly generated set of particles and uses the best particle based on an OF to update the velocity and position of the swarm [23].

GA and PSO are classical MH algorithms with similarities between them. Although the terminology differs—chromosomes in GA and particles in PSO—they both follow the same steps: generate a random set of solutions, evaluate them, and modify them iteratively until a stopping condition is met. Since most MHs follow a similar structure, new algorithms can be formalized as hybrid combinations of existing MHs or other AI paradigms [24].

A. MICRO EVOLUTIONARY PSO

MEPSO's main ideas are to reduce the original PSO's computational complexity and improve convergence time. The velocity component is replaced with crossover and mutation, while fewer particles are used in a micro population. This hybridization of the PSO algorithm takes advantage of a small population, achieving faster convergence, less computational complexity, and prevention of premature convergence, while using less memory to store the particles [20]. The algorithm consists of an inner loop that utilizes evolutionary operators to modify the particles and an external loop that controls the inner loop and determines the termination condition. While the external loop, based on the number of epochs defined, effectively controls the initialization of a population of particles; the inner loop, based on the number of iterations defined, updates each particle in a population following a specific algorithm with the evolutionary operators. Each loop is discussed in the following paragraphs.

The steps of MEPSO's external loop are defined in Figure 1. An initial population is generated with diversity ensured by the entropy concept from information theory; that is, particles must be sufficiently different from one another to proceed. Some steps are skipped in the first epoch, including the mutation of the best individual from the previous generation to include in the current population—a process known as elitism restart. After the inner loop completes, the algorithm checks termination criteria; if met, the process ends; otherwise, the epoch counter is incremented and a new population is generated following the initial steps. The stopping criteria include either a maximum number of epochs or a standard deviation close to zero over the last 10 epochs.

Each particle is modified in the inner loop with evolutionary operators following the algorithm proposed by [20], represented in Figure 2. Firstly, a particle with a solution $\mathbf{x} = (x_1, \ldots, x_n)$ from the population is mutated to $\mathbf{x}'_p$ by real number mutation using (1), based on a random number r between 0 and 1. If this number is greater than the hyperparameter μ, a solution value will remain the same; otherwise, the mutated value will be a product of the previous value, a random number between 0 and 1, and another hyperparameter σ. Both μ and σ are between 0 and 1 and they control, respectively, the frequency and magnitude of the mutations.

$$\forall x_i, \quad x'_i = \begin{cases} x_i, & \text{if } r > \mu \\ x_i \cdot \sigma \cdot \text{random}(0, 1), & \text{if } r \leq \mu \end{cases} \tag{1}$$

The next step is a crossover operator based on uniform crossover defined in (2). Two new offsprings y_1 and y_2 are generated from two parents (current particles) x_1 and x_2. Each offspring's solution value is changed using a random number δ_i between $-\gamma$ and $1 + \gamma$, which multiplies each parent's solution value. γ is another hyperparameter in MEPSO. Only the offspring with the best cost is selected. If a random number between 0 and 1 is less than a hyperparameter β, the current particle will undergo crossover with the global best (i.e., the best solution found so far). Otherwise, if it is less than a hyperparameter α, the algorithm computes the most dissimilar particle from the global best, based on Euclidean distance to its local best (i.e., the best of a single particle), and performs crossover between the current and most dissimilar particles. Both crossovers result in a new solution $\mathbf{x}_p^{\text{new}}$.

$$\begin{aligned} y_{1i} &= \delta_i x_{1i} + (1 - \delta_i) x_{2i} \\ y_{2i} &= (1 - \delta_i) x_{1i} + \delta_i x_{2i} \end{aligned} \tag{2}$$

Lastly, the current solution is updated to $\mathbf{x}'_p$ or $\mathbf{x}_p^{\text{new}}$ if they result in a better cost. The local best and global best are also updated. The inner loop is repeated for a predetermined number of iterations.

III. NEURO-FUZZY ALGORITHM

Neural networks can find relationships between variables after sufficient training with data, i.e., learning from samples, while a fuzzy inference system can solve complex problems that involve uncertainty. A neuro-fuzzy algorithm combines both approaches, and ANFIS is one of the most widely used among them. One of the advantages of ANFIS is its ability to explain its weights through a fuzzy inference system, whereas a traditional neural network typically cannot provide an interpretable explanation for its weight values [25]. The ANFIS structure with two inputs and two membership functions for each input is illustrated in Figure 3.

Five layers are responsible for the ANFIS functionality. The first layer uses membership functions to obtain fuzzy values. Specifically, a Gaussian function is calculated by

$$O_{1i} = \mu_{N_i}(a) = e^{-\frac{1}{2}\left(\frac{a-c}{d}\right)^2}, \quad i = 1, 2. \tag{3}$$

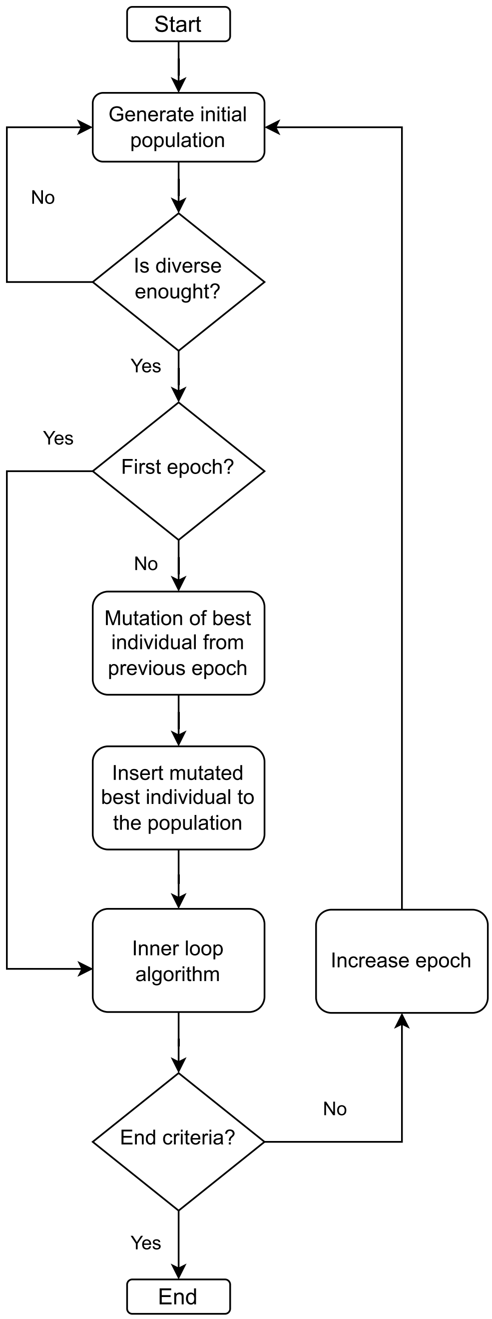


FIGURE 1. **MEPSO external loop algorithm flowchart.**

Resulting in four fuzzy values from inputs a and b, which are passed on to the next layer. Parameters c and d are called antecedent parameters and are defined through the training process. The second layer output is the product between two fuzzy values of different inputs to determine each rule firing strength w_i, using

$$O_{2i} = w_i = \mu_{A_i}(a) \cdot \mu_{B_i}(b), \quad i = 1, 2. \tag{4}$$

The normalization of each w_i is done in the third layer by dividing it by the sum of all rule firing strengths:

$$O_{3i} = \bar{w}_i = \frac{w_i}{w_1 + w_2 + w_3 + w_4}, \quad i = 1, 2, 3, 4. \tag{5}$$

Defuzzification happens in the fourth layer using a linear function that includes the original inputs with consequent parameters x_i, y_i, and z_i obtained by the training process:

$$O_{4i} = \bar{w}_i f_i = \bar{w}_i(x_i a + y_i b + z_i), \quad i = 1, 2, 3, 4. \tag{6}$$

Lastly, the ANFIS output is calculated in the last layer as

$$O_5 = \frac{\sum_i \bar{w}_i f_i}{\sum_i \bar{w}_i}. \tag{7}$$

Traditionally, the ANFIS algorithm uses a hybrid training method based on gradient descent and least squares estimation (LSE) to optimize, respectively, the antecedent and consequent parameters. However, there is potential to explore alternative approaches. Numerous studies in the field of TCM have applied ANFIS training based on meta-heuristic optimization [26]; therefore, the ANFIS-MEPSO model is proposed. Both antecedent and consequent parameters were optimized.

A. ANFIS-MEPSO

MEPSO optimization is employed to find the optimal values of the ANFIS model with the steps detailed in Figure 4. All ANFIS and MEPSO configurations are defined before the optimization process. Initially, the generated ANFIS model has standard parameters that are not specific to tool wear prediction; therefore, an optimization strategy is necessary to solve this problem. This is the same as the training phase of the ANFIS-MEPSO, because the goal of the MEPSO is to decrease the error between the ANFIS prediction and the real value of the output.

A population is initially generated following the diversity guidelines, with each particle containing the same number of values as the ANFIS' antecedent and consequent parameters. The elitism restart is employed after the first epoch to transfer the best particle from the previous epoch, using the mutation algorithm, to the current population.

The MEPSO's inner loop is executed, and the error of the current particle used as the ANFIS parameters in a prediction problem with the provided data defines the MEPSO cost. If the end criteria are not met, another initial population is generated, and the optimization loop continues.

After that, the best particle of all epochs is integrated into the ANFIS structure as its parameters, creating an optimized ANFIS-MEPSO model for tool wear prediction.

IV. METHODS

This section outlines the methodology of the study. The data used were processed through several methods to be applied in training and evaluating the ANFIS-MEPSO algorithm. A summary of the steps followed to achieve the results is included in Figure 5.

A. DATA PROCESSING

A public dataset containing sensor data from milling experiments, called the NASA milling dataset [27], was utilized in this work. Several sensor signals and experimental

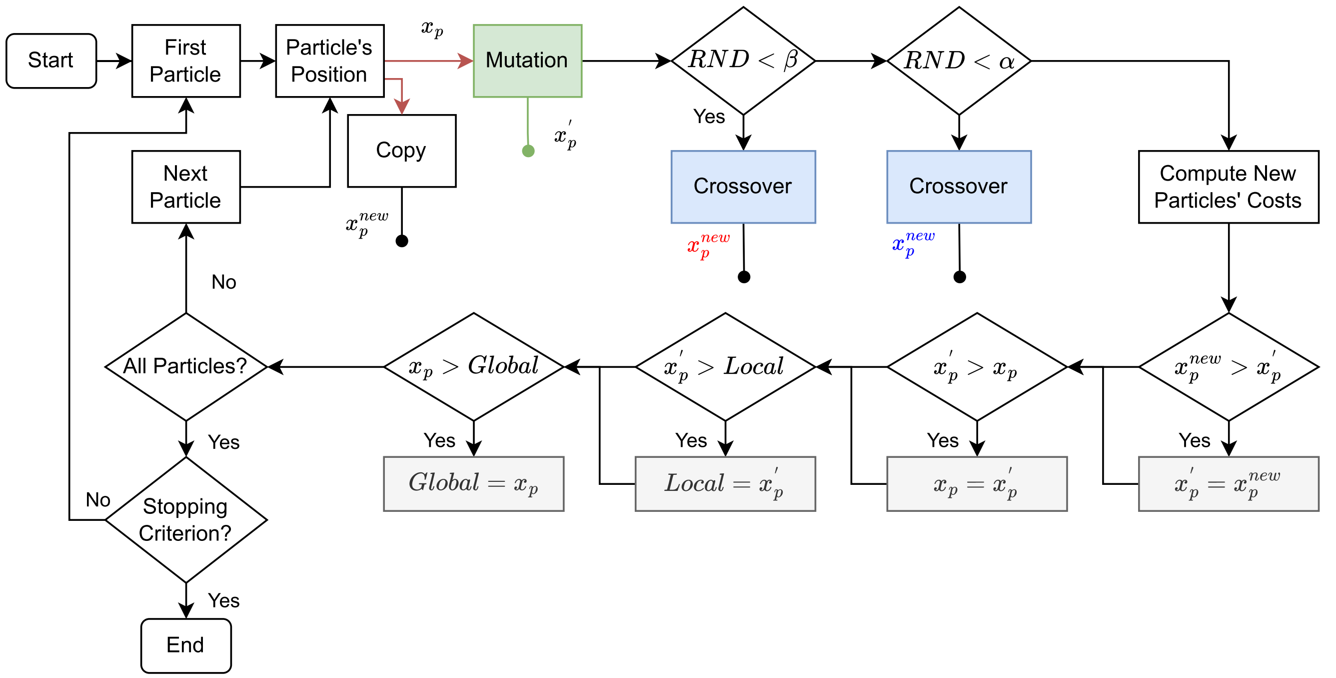


FIGURE 2. MEPSO inner loop algorithm flowchart.

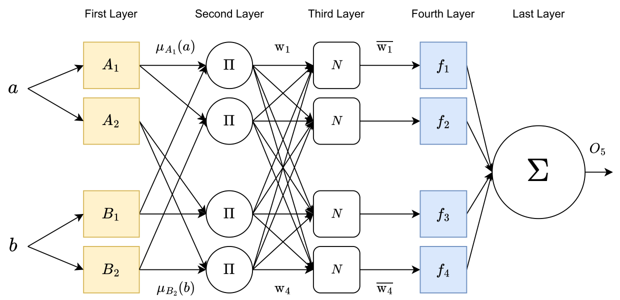


FIGURE 3. ANFIS structure with two inputs and two membership functions.

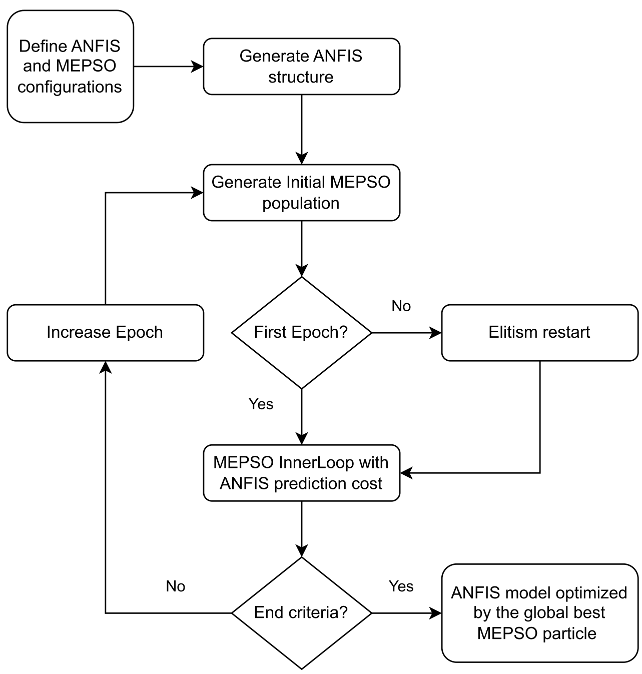


FIGURE 4. MEPSO's steps for optimizing ANFIS parameters.

parameter samples were collected in a MATLAB table, which was the main software used to process data and ANFIS-MEPSO algorithm training and testing.

The sensors provide vibration and AE signals from both the spindle and the table, as well as AC and DC electrical current signals from the motor, resulting in six unique types of data. Figure 6 displays sensor signals from a random milling experiment. Even though vibration, acoustic emission (AE), and current sensors have proven effective in capturing tool wear progression and condition in recent studies [28], [29], certain considerations must be taken into account. Vibration signals typically increase in amplitude as tool wear progresses; however, they are highly sensitive to variations in sensor mounting position and mechanical coupling. AE sensors are well established for detecting tool wear and breakage, yet their signals are often susceptible to environmental and structural noise. Current sensors also exhibit increasing signal amplitude as the spindle motor draws more current in response to higher cutting forces caused by tool blunting, though these signals can be affected by electrical noise and frictional fluctuations during machining.

The vibration and AE signals have amplitudes in root mean squared (RMS) form as a result of the dataset's data

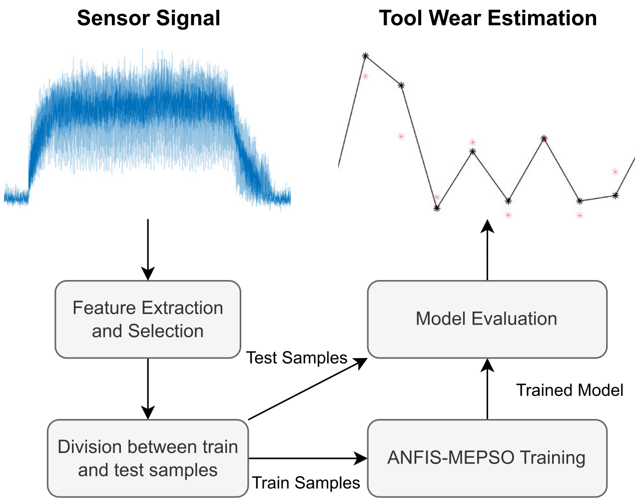


FIGURE 5. Summary of methods used in this work.

acquisition. Each signal contains 9000 measurement values for each tool pass over the workpiece, with a varying number of samples for each of the 16 experimental conditions, referred to as cases in the dataset. These conditions represent the experimental information and milling parameters. They are combinations of material (cast iron or steel), depth of cut (DOC) in mm, and feed rate in mm/rev used in the milling processes. All experiments were conducted employing a speed of 826 rev/min. A summary of all experiments is in Table 1. A total of 167 samples of milling experiments were acquired.

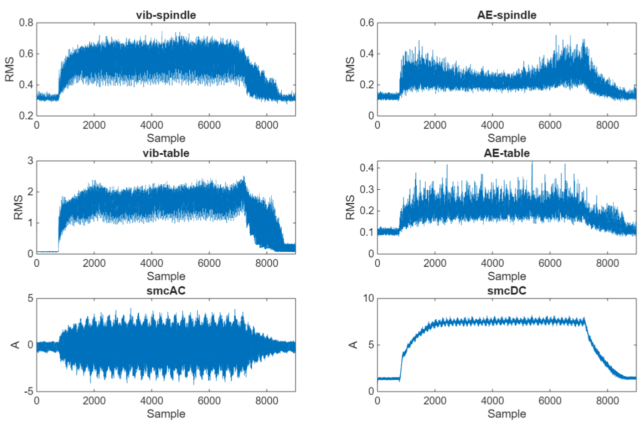


FIGURE 6. Sensor signals from a random milling experiment.

The tool wear is represented by the average variation in flank wear (VB) measured in mm. However, not all samples have measured values, and missing data were discarded as they cannot be used in supervised learning algorithms that require target values during training, nor can error metrics be calculated during testing; only 145 samples remained after that. The sensor signal's features were extracted in the time domain only, as shown in Table 2. The following

TABLE 1. Summary of milling experiments.

Case	DOC (mm)	Feed (mm/rev)	Material	Samples
1	1.5	0.5	Cast iron	17
2	0.75	0.5	Cast iron	14
3	0.75	0.25	Cast iron	16
4	1.5	0.25	Cast iron	7
5	1.5	0.5	Steel	6
6	1.5	0.25	Steel	1
7	0.75	0.25	Steel	8
8	0.75	0.5	Steel	6
9	1.5	0.5	Cast iron	9
10	1.5	0.25	Cast iron	10
11	0.75	0.25	Cast iron	23
12	0.75	0.5	Cast iron	15
13	0.75	0.25	Steel	15
14	0.75	0.5	Steel	10
15	1.5	0.25	Steel	7
16	1.5	0.5	Steel	6

notation is used: $x(n)$ represents the value of any signal, n is the sample number, N is the total number of samples in a signal, $\bar{x}$ is the signal mean, and σ is the signal standard deviation. Each sensor signal was used to extract its features, resulting in 48 different values. The selected features have been successfully employed in several studies on tool wear monitoring [13], [14], [15]. Frequency domain features were initially evaluated but did not provide relevant information in preliminary tests. This outcome can be attributed to the preprocessing applied to AE and vibration sensor signals, including filtering and RMS computation, which inherently suppress frequency-domain information by integrating signal energy over time. As a result, time domain features proved more effective in capturing relevant variations associated with tool wear progression.

B. FEATURE SELECTION

Several features were collected from the sensor signals. While some may be important for training the model, others will only add unnecessary complexity, based on the way that the ANFIS model adds more antecedent and consequent parameters with each input, without improving the performance. Therefore, a feature selection based on PCC was employed.

The PCC is a number that measures the correlation between two variables. This value r is between 0 and 1 for positive correlations. More information about this coefficient can be found in [15]. The feature extraction algorithm is represented in Figure 7. The time domain features in Table 2 were extracted from the sensor signals. The PCC of the feature and VB progression was then calculated, and the

TABLE 2. Time domain extracted features.

Feature	Formula
Different Peak	$I_1 = \max(x(n)) - \min(x(n))$
Peak	$I_2 = \max(\lvert x(n)\rvert)$
Crest Factor	$I_3 = \max\lvert x(n)\rvert / \sqrt{\frac{1}{N}\sum_{n=1}^{N} x(n)^2}$
Mean	$I_4 = \frac{1}{N}\sum_{n=1}^{N} x(n)$
Standard Deviation	$I_5 = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(x(n)-\bar{x})^2}$
Variance	$I_6 = \left(\frac{1}{N-1}\right)\sum_{n=1}^{N}(x(n)-\bar{x})^2$
Skewness	$I_7 = \left(\frac{1}{(N-1)\sigma^3}\right)\sum_{n=1}^{N}(x(n)-\bar{x})^3$
Kurtosis	$I_8 = \left(\frac{1}{(N-1)\sigma^4}\right)\sum_{n=1}^{N}(x(n)-\bar{x})^4$

values were organized in descending order. Lastly, the best and second best positive correlations were chosen to be part of the ANFIS-MEPSO inputs, namely P_4 and P_5. Milling parameters DOC, feed, and material, i.e., P_1, P_2, and P_3, were also considered, resulting in 5 unique features as inputs as shown in Figure 8, each with the same number of membership functions.

The number of features selected was chosen to reduce the computational complexity of the model, while retaining the best results. In this application, more inputs did not help the model perform better. That can be explained by understanding how the complexity works in the ANFIS model:

- The number of antecedent parameters can be calculated by the multiplication of the number of inputs, the number of membership functions and the number of parameters that can be modified in the membership function;
- The number of consequent parameters is related to the number of inputs that multiplies with the number of rules.

The number of all ANFIS parameters is the sum of the antecedent and consequent. We obtained 128 parameters that can be optimized with cluster influence range of 0.9 and the five chosen inputs. This number should be close to the number of samples of the training set to guarantee a model that is capable of generalization of unseen data.

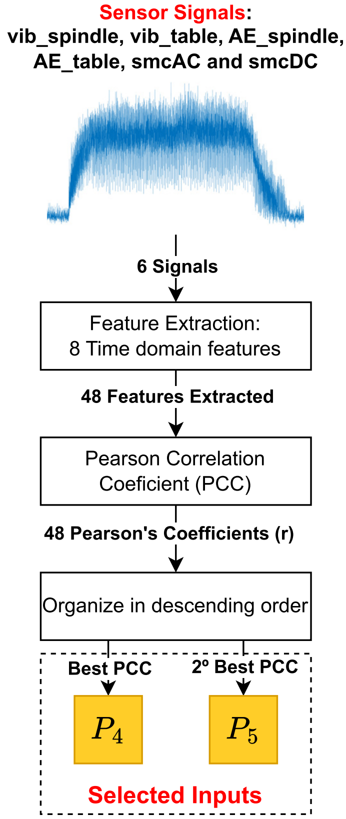


FIGURE 7. Feature selection algorithm.

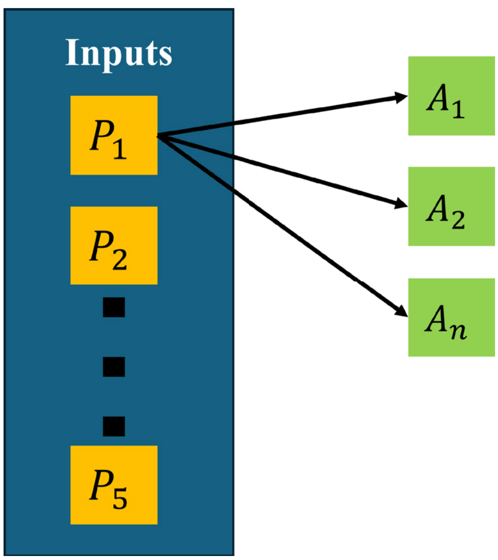


FIGURE 8. ANFIS-MEPSO's inputs.

C. DATA NORMALIZATION

Each input had a specific range of values. The milling experiment parameters, i.e. DOC, feed, and material, were composed of two distinct values, and the two features selected could have different ranges depending on the feature and sensor signal. Therefore, a normalization of data was conducted in each input and output to standardize the model's training and testing.

The normalization was made with the MATLAB function `normalize()`, using the "range" configuration, effectively modifying each input and output to a number between 0 and 1 considering its respective range.

D. ANFIS-MEPSO TRAINING AND TEST

The remaining 145 samples were divided into a training dataset and a test dataset. The split ratio was 90% of samples for training and 10% for testing, resulting in 130 samples to optimize the model parameters and 15 to test its performance in generalization after rounding the values. The specific samples in each set were chosen at random using the `randperm()` MATLAB function. The number generation was not fixed, resulting in different training and testing samples in each experiment.

An initial fuzzy system structure is generated by the `genfis()` MATLAB function with the Sub-Clustering option and Gaussian membership functions. Most of the function's hyperparameters were set to standard values, except the cluster influence range, which was changed to 0.9. The structure of the ANFIS model is defined at this point, specifically the number of membership functions, rules, and parameters.

The train parameters of the ANFIS model are optimized using the MEPSO algorithm. The root-mean-squared error (RMSE) metric between the actual normalized value and the ANFIS output was used as the MEPSO cost. Thus, the ANFIS-MEPSO model adjusts its parameters to achieve the lowest possible error over a given number of epochs. All MEPSO hyperparameters are listed in Table 3. Most of them are optimal hyperparameters found by [20] after a series of experiments to determine the best possible combination in benchmark tests. Only the number of iterations and epochs were changed based on the ANFIS-MEPSO experiments in an attempt to reduce the overall computational complexity while still producing good results. The particle solution values are randomly generated in the initial population but were limited in a specific way. Antecedent values correspond to the two Gaussian membership function parameters; hence, the random numbers had a lower bound of 0.1 and 0, and the upper bound was chosen based on the fuzzy range (FR), i.e., limits of the number input in the fuzzy membership function, of the initial generated fuzzy system, specifically an upper bound of FR/2 and FR. Consequent parameters had bounds of −10 and 10 for every value. These limitations were defined heuristically, and they ensured that the algorithm generates reasonable initial values that match the ANFIS' particularities.

All tests were performed on the remaining 10% of the samples. The R^2 and mean absolute error (MAE) metrics were employed to evaluate the ANFIS-MEPSO model's capability to generalize the data with minimal error. All results were limited to values between 0 and 1 before computing the metrics, since the model is expected to estimate values within the normalization range. The results were also denormalized for some illustrations in the results section.

TABLE 3. MEPSO's hyperparameters.

Hyperparameter	Value
β	0.6
α	0.9
μ	0.5
γ	0.7
Number of particles	25
Iterations	20
Epochs	5

V. RESULTS

This section presents the ANFIS-MEPSO results in tool wear estimation using milling experimental data.

A. OPTIMIZATION RESULTS

The MEPSO optimization in the training phase of the ANFIS model was conducted several times with the aforementioned hyperparameters in Table 3. The computer unit used in the training part was the Intel Core Ultra 9 185H with 5.1 GHz of speed, also considering no graphical processing or multi-core computing. The training phase took 3.11 hours to complete in 5 epochs with 20 iterations each, averaging in 0.622 hours or 37.32 minutes per epoch. The trained model calculated the results in 15 test samples with an inference time of 0.031383 seconds, averaging 2.0922 milliseconds for each sample.

The inference time of the trained model showed promising results for deployment in milling applications, specially after some tweaking with the hardware and code. However, the training time per epoch was subpar for online optimization applications in the tested configuration. It is important to note that the computing times could be improved by multi-core or graphics card processing.

B. NUMERICAL RESULTS IN TESTING DATA

Testing data is crucial to assess the model's ability to generalize to unseen samples; therefore, these results directly reflect the performance of the ANFIS-MEPSO model. Table 4 shows the measured and predicted VB tool wear on the testing set, along with the prediction error for each sample. Considering that all VB values range from 0 to 1.4 mm, most predictions were accurate within an absolute error margin of less than 0.1 mm. Some outliers, such as samples 4 and 15, exhibited greater errors, indicating difficulties in predicting null values. Most prediction errors were negative, meaning the model tended to overestimate the tool wear, which suggests a tendency of the model to produce higher-than-actual predictions.

TABLE 4. Tool wear estimation on testing data.

Test Sample	Measured VB (mm)	Predicted VB (mm)	Error (mm)
1	0.1200	0.1027	0.0173
2	0.2800	0.3315	-0.0515
3	0.0500	0.0000	0.0500
4	0.0000	0.1251	-0.1251
5	0.5800	0.5480	0.0320
6	0.5500	0.6173	-0.0673
7	0.2600	0.2912	-0.0312
8	0.2700	0.3028	-0.0328
9	0.8100	0.7340	0.0760
10	0.4800	0.5682	-0.0882
11	0.2300	0.2704	-0.0404
12	0.1000	0.1861	-0.0861
13	0.4600	0.4067	0.0533
14	0.3000	0.3012	-0.0012
15	0.1500	0.2642	-0.1142

Numerical metrics, namely MSE, MAE, and R^2, based on the error of ANFIS-MEPSO predictions for training, testing, and all data are compiled in Table 5. The results display low MSE and MAE values for the test set, but higher errors on the training samples, suggesting a strong generalization capability of the model on unseen data. The R^2 metric for the test samples also achieved a high value, reinforcing the model's performance in the tool wear estimation task, especially considering that $R^2 = 1$ represents the best possible outcome.

TABLE 5. ANFIS-MEPSO result metrics.

Data Group	MSE (mm^2)	MAE (mm)	R^2
Train	0.0131	0.0828	0.8113
Test	0.0045	0.0578	0.9058
All	0.0122	0.0802	0.8185

A visualization of the measured and predicted VB by the ANFIS-MEPSO model on the testing data is shown in Figure 9. The solid line represents perfect prediction; thus, the closer the points are to this line, the more accurate the prediction. VB values between 0.2 and 0.3 mm yielded the most accurate results, generally falling slightly above the measured values. Predictions for VB values below 0.2 mm showed less accuracy overall. For values above 0.4 mm, the absolute prediction error was consistently below 0.1 mm, demonstrating the ANFIS-MEPSO model's robustness and accuracy in estimating higher tool wear values.

Figure 10 shows the VB values estimated by the ANFIS-MEPSO model in the test data. All samples were randomly

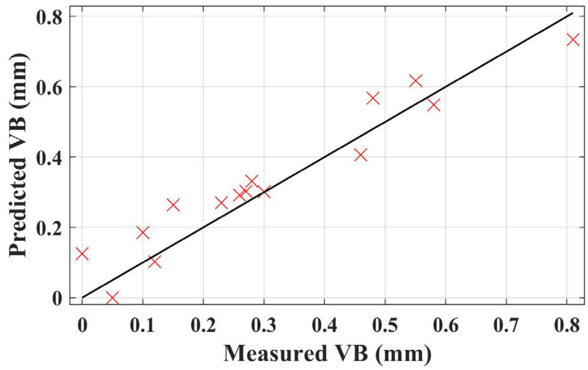


FIGURE 9. Scatter plot of measured and predicted VB by ANFIS-MEPSO model.

selected, and their VB values were below 1.4 mm. The accuracy of the model is further confirmed across most samples, as reflected by the evaluation metrics and the measured versus predicted values illustrated.

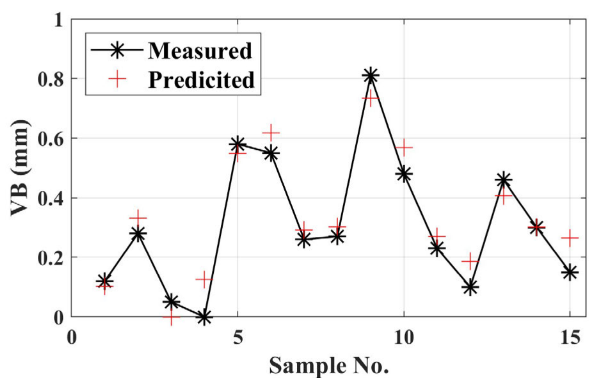


FIGURE 10. Regression plot of measured and predicted VB by ANFIS-MEPSO model.

C. COMPARATIVE ANALYSIS WITH SIMILAR MODELS

Similar models have been used to estimate tool wear in milling processes; therefore, a comparison between these approaches and the proposed method is provided in Table 6, whereas the corresponding metrics are compiled in Table 7. Most studies employed SVM and ANFIS models with MH optimization based on GWO, WOA, or PSO, while [13] and [16] proposed an ELM model with GA-PSO and an SVR model with DE-GWO, respectively. This work presented an ANFIS model with MEPSO optimization, making it most similar to the approach in [19]. Other key differences between the methods include the types of sensor signals used, feature extraction domains (single or multiple), and whether feature selection was applied. Additionally, the datasets used for validation and performance metrics—such as MSE, RMSE, MAE, mean absolute percent error (MAPE), and R^2—also vary across studies. The proposed work is the only one to use the NASA milling dataset, while others rely on experimental or PHM Data Challenge datasets. Although the evaluation

TABLE 6. Comparison between similar approaches and the proposed method.

Work	Model	Feature Extraction and Selection	Validation Methodology and Metrics
[13]	ELM with GA and PSO optimization	Multi-domain feature extraction of vibration signals	PHM open-access dataset and experimental data with MSE
[14]	SVM and GWO optimization	Multi-domain feature extraction of cutting force signals with GA based selection	Experimental data with accuracy on classification of tool wear state
[15]	SVM with WOA optimization	Multi-domain feature extraction of multiple sensor signals with PCC based selection	PHM open-access dataset with MAE and RMSE
[16]	SVR with DE and GWO optimization	Multi-domain feature extraction of multiple sensor signals with PCC and F-score based selection	Experimental data with MAE, MAPE, RMSE and R^2
[19]	ANFIS with vibration based PSO	Cutting force and cutting parameters without feature extraction or selection	Experimental data with MSE, MAPE and R^2
Present work	ANFIS with MEPSO optimization	Time-domain feature extraction of multiple sensor signals with PCC based selection	NASA milling dataset [27] with MSE, MAE and R^2

metrics differ among the studies, preventing a fully direct comparison, an assessment based on the shared metrics used to validate the ANFIS-MEPSO model remains viable.

The present work's 0.0045 mm^2 MSE metric achieved a better result than [19], which obtained 0.0060 mm^2 using a similar ANFIS-based model. In comparison, the study in [13] achieved an average of 89.5267 μm^2 on the PHM dataset. However, since their experimental data considered tool wear area, this result cannot be directly compared. Considering that the MSE metric involves squared units, a division by 1,000,000 is required to convert μm^2 to mm^2. This conversion reveals that the MSE reported in [13] is numerically lower, although based on a different wear representation. Nevertheless, the ANFIS-MEPSO model achieved a better MSE result when compared to a similar ANFIS model.

The MAE obtained was 0.0578 mm, whereas the results in [15] averaged 0.0075 mm using experimental data, outperforming the ANFIS-MEPSO model. Meanwhile, the results in [16] reached 36 μm on the PHM dataset, equivalent to 0.0036 mm, thus achieving the lowest error among the three studies.

TABLE 7. Compiled similar approaches comparison metrics.

Work	MSE (mm^2)	MAE (mm)	R^2
[13]	**89.527 μm^2**	-	-
[14]	-	-	-
[15]	-	0.0075	-
[16]	-	**0.0036**	**0.9634**
[19]	0.0060	-	0.9540
Present work	0.0045	0.0578	0.9058

D. COMPARATIVE ANALYSIS WITHIN THE SAME DATASET

Several works relied on the same milling experimental dataset as their data source for tool wear estimation. A comparison between these studies and the proposed method is presented in Table 8. Some of the works evaluated multiple machine learning algorithms for performance benchmarking; however, only the main proposed model or the model with the best results was considered in this comparison. As seen in Table 9, the evaluation metrics differ across studies, which adds another level of complexity to a direct comparison.

TABLE 8. Comparison with other approaches and the proposed method within the same dataset.

Work	Model	Feature Extraction and Selection	Validation Methodology and Metrics
[5]	ANN	Multi-domain feature extraction of multiple sensor signals	MSE and R metrics
[30]	Enhanced ELM	Multi-domain feature extraction of electrical current signals	MAE, RMSE and R metrics
[31]	LSTM network	LSTM feature extraction	MAE and RMSE metrics
Present work	ANFIS with MEPSO optimization	Time-domain feature extraction of multiple sensor signals with PCC based selection	MSE, MAE and R^2 metrics

TABLE 9. Compiled same dataset approaches comparison metrics.

Work	MSE (mm^2)	MAE (mm)	R^2
[5]	**0.0044**	-	0.9128
[30]	-	**0.0134**	**0.9934**
[31]	-	0.0322	-
Present work	0.0045	0.0578	0.9058

The study by [5] applied SVM, ANN, and GPR models to predict tool wear values. The best performance was achieved using a neural network with multi-domain feature extraction from all available sensors in the dataset. Process parameters such as DOC and material type were also included in the input vector, similarly to the ANFIS-MEPSO training approach. Their model achieved a MSE of 0.0044 mm^2, which is very close to the 0.0045 mm^2 MSE obtained by the proposed model under standard optimization. Regarding the R^2 metric, the ANFIS-MEPSO reached 0.9058, while [5] reported an R value of 0.9554, corresponding to an R^2 of 0.9128, slightly higher than that of the proposed method.

In another study, [30] implemented an enhanced ELM using only electrical current signals, with multi-domain feature extraction. Their model achieved a MAE of 0.0134 mm, outperforming the MAE of 0.0578 mm obtained by the ANFIS-MEPSO model. Additionally, they reported an R value of 0.9967, resulting in an R^2 of 0.9934, which is considerably superior to the R^2 of 0.9058 from the proposed method.

Lastly, [31] used long short-term memory (LSTM) networks, benefiting from the model's ability to automatically extract features from the raw signals. Their model achieved a MAE of 0.0322 mm, which is better than the 0.0578 mm obtained with the standard optimization of ANFIS-MEPSO.

All the studies mentioned above used the same publicly available dataset to validate their models. However, their approaches differ significantly in terms of feature extraction. Most of them employed multi-domain feature extraction (time, frequency, and time-frequency), and neural network-based models that allow for a larger number of input variables without the use of feature selection algorithms. Moreover, the train-test splits used in each study are not identical, which further complicates a direct and fair performance comparison across models.

E. LOCAL COMPARATIVE ANALYSIS WITH ANFIS MODELS

To understand the ANFIS-MEPSO performance gains in regards to other similar approaches in optimization, a local comparison with similar models was conducted with the same training and test datasets. The same evaluation internal metrics (MAE, MSE and R^2) used in the previous analyses were adopted, and baseline models were implemented to measure the impact of MEPSO on the ANFIS performance. Specifically, the ANFIS-MEPSO approach, the original ANFIS with hybrid training, and other metaheuristics, i.e. ANFIS-GA and ANFIS-PSO, were tasked to estimate tool wear in an identical setting. The other approaches were trained with the updated `tunefis()` MATLAB function that allows for classical and alternative optimization methods to tune the ANFIS parameters. Hyperparameters of these optimizers and the original ANFIS with hybrid training were defined through a trial and error effort to achieve the best possible results, which are compiled in Table 10. Only the modified hyperparameters in the trial and error were included, while the others were left in standard MATLAB values.

The results are compiled in Table 11. Overall, the results produced by the models with any metaheuristic optimization performed better than the classical hybrid learning approach. The ANFIS R^2 metric presented 0.7331 against R^2 metrics of at least 0.8790 in the other models, also considering greater MSE and MAE metrics. The ANFIS-MEPSO model presented better R^2 and MSE metrics of, respectively, 0.9058 and 0.0045, compared with GA and PSO, although the GA was a close second. The best MAE metric of 0.0530 was from the PSO optimizer with a significant advantage over MEPSO and GA, but its R^2 metric ranked third and did not reach the 0.9 threshold.

TABLE 10. Baseline models' MATLAB hyperparameters.

Model	Hyperparameter	Value
ANFIS	EpochNumber	10,000
ANFIS-GA	CreationFcn	'gacreationuniform'
	MutationFcn	'mutationadaptfeasible'
	CrossoverFcn	'crossoverscattered'
	SelectionFcn	'selectionstochunif'
	CrossoverFraction	0.8
	EliteCount	10
ANFIS-PSO	SwarmSize	100
	SelfAdjustmentWeight	1.49
	SocialAdjustmentWeight	1.49

TABLE 11. Compiled local similar models comparison metrics.

Model	MSE (mm^2)	MAE (mm)	$\mathbf{R^2}$
ANFIS	0.0127	0.0717	0.7331
ANFIS-GA	0.0046	0.0568	0.9029
ANFIS-PSO	0.0057	**0.0530**	0.8790
ANFIS-MEPSO	**0.0045**	0.0578	**0.9058**

All of the models utilized the same data; therefore, it was a fairer comparison. A metaheuristic approach on the optimization of ANFIS in the tool wear estimation context showed improvements on the more traditional method. The GA, PSO, and MEPSO algorithms produced good results with similar metrics. While the MEPSO gave a smaller MSE and biggest R^2, it also produced the worst MAE of the three metaheuristics comparatively.

F. SENSITIVITY ANALYSIS

To evaluate the robustness of the proposed approach, a sensitivity analysis was performed. This analysis examines how variations in the model input parameters influence the output, primarily addressing the uncertainty of model predictions with respect to the inputs [32]. In addition to numerical inputs, small modifications in the model structure and training process are also relevant for assessing generalization to unseen data, as noted in [33]. In this study, such variations were introduced in both the ANFIS structure and the MEPSO optimization. Altering the number of ANFIS parameters by increasing the number of fuzzy rules and thereby raising the model's complexity proved ineffective during the standard training and testing phases. Consequently, the evaluation of output variability and performance focused on adjusting the hyperparameters of the MEPSO optimization, not only by increasing training complexity but also by modifying the influence of the evolutionary operators.

To determine exactly how the changes in the MEPSO optimization affect the output, the number of particles, σ and γ were chosen to represent the analysis. Changing the population size alters the number of chances that the algorithm can find the global minimum, while it also increases the time it takes to train. σ and γ directly affect the mutation and crossover operators, changing how much they are relevant in the optimization. Table 12 refers to the experiments that were conducted on the same data from prior sections three times to mitigate the stochastic effect of the MEPSO optimization.

TABLE 12. Combination of sensitivity experiments.

Hyperparameter	Values	Experiments
Number of Particles	[10, 15, 20, 25, 30]	5
σ	[0.6, 0.7, 0.8, 0.9]	4
γ	[0.6, 0.7, 0.8, 0.9]	4
Total Number of Experiments		240

The R^2 metric was chosen to represent the changes in performance. Only the best metric out of the three tries was utilized. Since we are modifying three hyperparameters, the σ was fixed in each representation of a bar graph with the number of particles as the x-axis and different colored bars as variations in γ.

The Figure 11 represents the changes in performance with a σ fixed in 0.6. Firstly, the effect of the number of particles is not straightforward; more particles do not necessarily mean better performance, as it can be noted with 15 particles that γ with 0.7 and 0.8 produced excellent performance while 0.9 had a subpar result. The γ changes are relevant in this case, considering the 0.7 and 0.8 produced better results with fewer particles. While the combination of $\gamma = 0.7$ and 15 particles has the best result, 0.8 was a close second and kept the excellent performance in any number of particles.

The metric variations in Figure 12 represent combinations that had a $\sigma = 0.7$. The results with a $\gamma = 0.6$ produced less stable results than the previous ones. Most of them range from an R^2 of 0.7 and 0.9, except with 20 particles that had poor performance, but considering the highly stochastic behavior of the algorithm, it could be only a lack of a few more tries. The next representation in Figure 13 has a fixed $\sigma = 0.8$. The best results were obtained in $\gamma = 0.6, 0.7, 0.8$ with 20 and 25 particles, while $\gamma = 0.9$ had less consistent results.

Finally, the last set of variations in Figure 14 is less consistent than the previous one, notably with fewer particles. Considering that the number of particles is tied to the

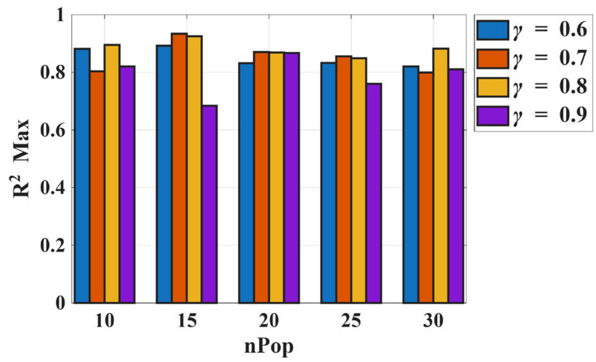


FIGURE 11. R^2 variation on multiple γ values and number of particles with a fixed $\sigma = 0.6$.

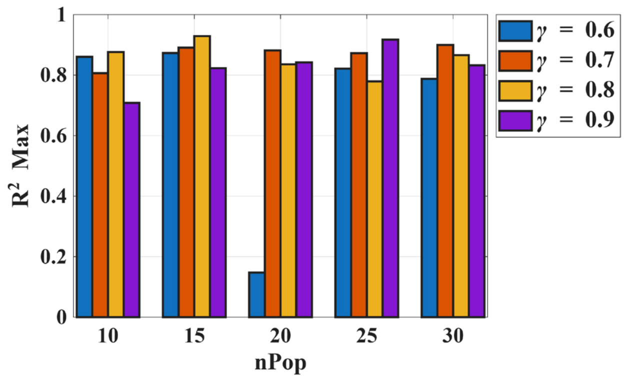


FIGURE 12. R^2 variation on multiple γ values and number of particles with a fixed $\sigma = 0.7$.

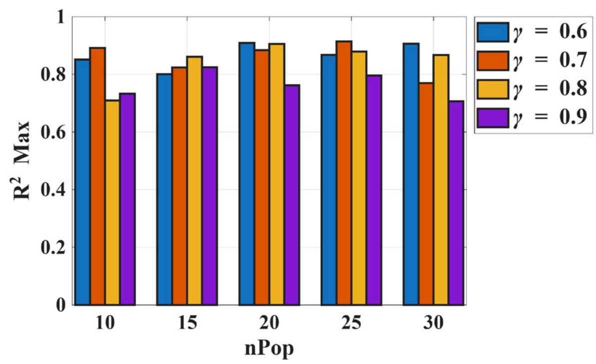


FIGURE 13. R^2 variation on multiple γ values and number of particles with a fixed $\sigma = 0.8$.

optimization complexity, it can be said, excluding a few outliers, that the results from 20 to 30 particles are usually more consistent than 10 to 15, as noted in $\sigma = 0.6, 0.7$. The optimal hyperparameter combination as stated in [20] produced reliable results in any number of particles. The results at $\sigma = 0.9$ were overall less consistent with worst performance, similar to what happens at $\gamma = 0.9$, suggesting that these combinations are more sensitive to a specific number of particles. The best combinations involved σ and

γ with a value between 0.6 and 0.8 and usually at least 20 particles.

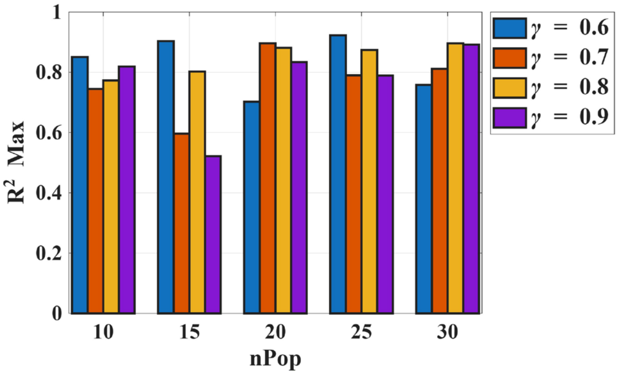


FIGURE 14. R^2 **variation on multiple** γ **values and number of particles with a fixed** $\sigma = 0.9$.

VI. CONCLUSION

This work presented a concise approach to tool wear estimation by employing a hybrid model, ANFIS-MEPSO. The model achieved accurate predictions, as validated by performance metrics including MSE, MAE, and R^2. These promising results demonstrate the applicability of the proposed model, particularly in milling scenarios. In addition, the study addressed essential steps in TCM-related problems, such as sensor signal feature extraction and selection, adjustments to the MEPSO optimization technique to better tune the antecedent and consequent parameters of the ANFIS model, and a sensitivity analysis based on the variation of evolutionary hyperparameters to better understand its impact on performance.

Specifically, the model achieved a MSE of 0.0045, a MAE of 0.0578, and an R^2 of 0.9058 on the test data. When compared to similar approaches in the literature, the ANFIS-MEPSO model produced results that are either comparable or superior in terms of MSE. These findings underscore the robustness and effectiveness of the proposed method for tool wear estimation, making it a valuable addition to existing research in this area. The sensitivity analysis demonstrated that performance is strongly influenced by the selection of evolutionary hyperparameters and the number of particles. Configurations with σ and γ between 0.6 and 0.7 consistently achieved more stable results, particularly when using at least 20 particles. In contrast, combinations with $\sigma = 0.9$ or $\gamma = 0.9$ tended to produce less consistent outcomes, highlighting their sensitivity to the optimization complexity.

Future work may include applying ANFIS-MEPSO to a broader range of experimental data with diverse milling conditions or using other publicly available datasets. Moreover, hybrid training strategies could be explored—where MEPSO is used to optimize antecedent parameters and a mathematical method such as least squares estimation (LSE) is applied to the consequent parameters—to further enhance model performance. A study on the feasibility of the edge deployment and in-line control of the algorithm on a milling process would be interesting to explore its capabilities in an industry application. The ANFIS-MEPSO could also be applied in different machining processes with similar monitoring techniques.

REFERENCES

[1] B. Sener, G. Serin, M. U. Gudelek, A. M. Ozbayoglu, and H. O. Unver, "Intelligent chatter detection in milling using vibration data features and deep multi-layer perceptron," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2020, pp. 4759–4768.

[2] S. Shankar, T. Mohanraj, and R. Rajasekar, "Prediction of cutting tool wear during milling process using artificial intelligence techniques," *Int. J. Comput. Integr. Manuf.*, vol. 32, no. 2, pp. 174–182, Feb. 2019.

[3] M. Iliyas Ahmad, Y. Yusof, M. E. Daud, K. Latiff, A. Z. Abdul Kadir, and Y. Saif, "Machine monitoring system: A decade in review," *Int. J. Adv. Manuf. Technol.*, vol. 108, nos. 11–12, pp. 3645–3659, Jun. 2020.

[4] M. Wiciak-Pikuła, A. Felusiak-Czyryca, and P. Twardowski, "Tool wear prediction based on artificial neural network during aluminum matrix composite milling," *Sensors*, vol. 20, no. 20, p. 5798, Oct. 2020.

[5] Y. E. Karabacak, "Intelligent milling tool wear estimation based on machine learning algorithms," *J. Mech. Sci. Technol.*, vol. 38, no. 2, pp. 835–850, Feb. 2024.

[6] Y. Feng, "A holistic approach for improving milling machine cutting tool wear prediction," *Appl. Intell.*, vol. 53, no. 24, pp. 30329–30342, Dec. 2023.

[7] C. Shi, B. Luo, S. He, K. Li, H. Liu, and B. Li, "Tool wear prediction via multidimensional stacked sparse autoencoders with feature fusion," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5150–5159, Aug. 2020.

[8] Z. He, T. Shi, and J. Xuan, "Milling tool wear prediction using multi-sensor feature fusion based on stacked sparse autoencoders," *Measurement*, vol. 190, Feb. 2022, Art. no. 110719.

[9] Z. He, T. Shi, and X. Chen, "An innovative study for tool wear prediction based on stacked sparse autoencoder and ensemble learning strategy," *Sensors*, vol. 25, no. 8, p. 2391, Apr. 2025.

[10] K. Zhu, H. Guo, S. Li, and X. Lin, "Physics-informed deep learning for tool wear monitoring," *IEEE Trans. Ind. Informat.*, vol. 20, no. 1, pp. 524–533, Jan. 2024.

[11] S. Li, J. Li, and K. Zhu, "Application of physics-guided deep learning model in tool wear monitoring of high-speed milling," *Mech. Syst. Signal Process.*, vol. 224, Feb. 2025, Art. no. 111949.

[12] A. Jamali, A. Kashyap, J. Schneider, M. Stueber, and V. Schulze, "Grey-box modelling for tool wear prediction in milling: Fusion of finite element insights, time-resolved cutting signals and metaheuristic feature selection," *Wear*, vols. 580–581, Oct. 2025, Art. no. 206292.

[13] Z. Lei, Q. Zhu, Y. Zhou, B. Sun, W. Sun, and X. Pan, "A GAPSO-enhanced extreme learning machine method for tool wear estimation in milling processes based on vibration signals," *Int. J. Precis. Eng. Manuf.-Green Technol.*, vol. 8, no. 3, pp. 745–759, May 2021.

[14] X. Liao, G. Zhou, Z. Zhang, J. Lu, and J. Ma, "Tool wear state recognition based on GWO-SVM with feature selection of genetic algorithm," *Int. J. Adv. Manuf. Technol.*, vol. 104, nos. 1–4, pp. 1051–1063, Sep. 2019.

[15] Y. Cheng, X. Gai, Y. Jin, R. Guan, M. Lu, and Y. Ding, "A new method based on a WOA-optimized support vector machine to predict the tool wear," *Int. J. Adv. Manuf. Technol.*, vol. 121, nos. 9–10, pp. 6439–6452, Aug. 2022.

[16] J. Wang, H. Liu, X. Qi, Y. Wang, W. Ma, and S. Zhang, "Tool wear prediction based on SVR optimized by hybrid differential evolution and grey wolf optimization algorithms," *CIRP J. Manuf. Sci. Technol.*, vol. 55, pp. 129–140, Dec. 2024.

[17] K. Tan, Z. Liu, and R. Jiang, "Intelligent tool-wear prediction for ball nose tungsten carbide cutters in milling of stainless steel HRC52 using a genetic algorithm-enhanced bi-directional long short-term memory network," *Discover Appl. Sci.*, vol. 7, no. 10, Oct. 2025, Art. no. 1166. [Online]. Available: https://doi.org/10.1007/s42452-025-07007-z

[18] Z. Che, C. Peng, C. Wang, and J. Wang, "A novel integrated TDLAVOA-XGBoost model for tool wear prediction in lathe and milling operations," *Results Eng.*, vol. 27, Sep. 2025, Art. no. 105984.

[19] L. Xu, C. Huang, C. Li, J. Wang, H. Liu, and X. Wang, "Estimation of tool wear and optimization of cutting parameters based on novel ANFIS-PSO method toward intelligent machining," *J. Intell. Manuf.*, vol. 32, no. 1, pp. 77–90, Jan. 2021.

[20] B. J. Solano-Rojas, R. Villalón-Fonseca, and R. Batres, "Micro evolutionary particle swarm optimization (MEPSO): A new modified metaheuristic," *Syst. Soft Comput.*, vol. 5, Dec. 2023, Art. no. 200057.
[21] R. Salgotra, P. Sharma, S. Raju, and A. H. Gandomi, "A contemporary systematic review on metaheuristic optimization algorithms with their MATLAB and Python code reference," *Arch. Comput. Methods Eng.*, vol. 31, no. 3, pp. 1749–1822, Apr. 2024.
[22] A. Slowik and H. Kwasnicka, "Evolutionary algorithms and their applications to engineering problems," *Neural Comput. Appl.*, vol. 32, no. 16, pp. 12363–12379, Aug. 2020.
[23] J. Tang, G. Liu, and Q. Pan, "A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 10, pp. 1627–1643, Oct. 2021.
[24] A. Seyyedabbasi, R. Aliyev, F. Kiani, M. U. Gulle, H. Basyildiz, and M. A. Shah, "Hybrid algorithms based on combining reinforcement learning and metaheuristic methods to solve global optimization problems," *Knowledge-Based Syst.*, vol. 223, Jul. 2021, Art. no. 107044.
[25] D. Karaboga and E. Kaya, "Adaptive network based fuzzy inference system (ANFIS) training approaches: A comprehensive survey," *Artif. Intell. Rev.*, vol. 52, no. 4, pp. 2263–2293, Dec. 2019.
[26] L. Xu, C. Huang, C. Li, J. Wang, H. Liu, and X. Wang, "A novel intelligent reasoning system to estimate energy consumption and optimize cutting parameters toward sustainable machining," *J. Cleaner Prod.*, vol. 261, Jul. 2020, Art. no. 121160.
[27] A. Agogino and K. Goebel, "Milling data set. NASA AMES prognostics data repository," NASA Ames Research Center, Moffett Field, CA, USA, Tech. Rep. 3, 2007. [Online]. Available: https://data.phmsociety.org/nasa/
[28] S. Sayyad, S. Kumar, A. Bongale, P. Kamat, S. Patil, and K. Kotecha, "Data-driven remaining useful life estimation for milling process: Sensors, algorithms, datasets, and future directions," *IEEE Access*, vol. 9, pp. 110255–110286, 2021.
[29] A. Mohamed, M. Hassan, R. M'Saoubi, and H. Attia, "Tool condition monitoring for high-performance machining systems—A review," *Sensors*, vol. 22, no. 6, p. 2206, Mar. 2022.
[30] Y. Zhou and W. Sun, "Tool wear condition monitoring in milling process based on current sensors," *IEEE Access*, vol. 8, pp. 95491–95502, 2020.
[31] W. Cai, W. Zhang, X. Hu, and Y. Liu, "A hybrid information model based on long short-term memory network for tool condition monitoring," *J. Intell. Manuf.*, vol. 31, no. 6, pp. 1497–1510, Aug. 2020.
[32] M. Safari, V. Tahmasbi, and A. H. Rabiee, "Investigation into the automatic drilling of cortical bones using ANFIS-PSO and sensitivity analysis," *Neural Comput. Appl.*, vol. 33, no. 23, pp. 16499–16517, Dec. 2021.
[33] H. Moayedi, M. Raftari, A. Sharifi, W. A. W. Jusoh, and A. S. A. Rashid, "Optimization of ANFIS with GA and PSO estimating α ratio in driven piles," *Eng. Comput.*, vol. 36, no. 1, pp. 227–238, Jan. 2020.

GIOVANNI O. DE SOUSA received the B.S. degree in controls and automation engineering from the Federal Institute of Amazonas, Manaus, Brazil, in 2022, and the M.S. degree in electrical engineering from the University of São Paulo, São Carlos, Brazil, in 2025, where he is currently pursuing the Ph.D. degree.

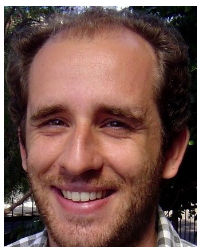
He is actively involved in machine learning and intelligent manufacturing research.

PAULO M. C. MONSON received the bachelor's degree in industrial automation from the Faculty of Technology, São Paulo, Brazil, in 2021, and the M.Sc. degree in electrical engineering from São Paulo State University, Bauru, Brazil, in 2023. He is currently pursuing the Ph.D. degree in electrical engineering with the University of São Paulo, São Carlos, Brazil.

He is also a Researcher with the DataSense–Smart Sensing and Data Analysis Research Group. He has experience in the field of digital signal processing, computer vision, and deep learning.

FÁBIO R. L. DOTTO was born in Bauru, São Paulo, Brazil, in 1977. He received the M.Sc. degree in industrial engineering and the Ph.D. degree in electrical engineering both from São Paulo State University (UNESP), São Paulo, in 2004 and 2019, respectively.

He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, São Carlos School of Engineering, University of São Paulo (USP), São Paulo. His research interests include instrumentation, intelligent sensors, data acquisition, fault diagnosis, and the Industrial Internet of Things (IIoT) applications for Industry 4.0.



DENNIS BRANDAO (Member, IEEE) received the Ph.D. degree from the University of São Paulo (USP), São Carlos, Brazil.

He was with Electronic Research and Development for four years and also a Field Service Engineer for industrial automation companies. He is currently an Associate Professor with the Department of Information Engineering, University of Brescia, Brescia, Italy. He is a member of the National ABNT/CB-03/TC-65 Committee of ISO.

PEDRO O. C. JUNIOR (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in electrical engineering from São Paulo State University, Bauru, Brazil, in 2016 and 2020, respectively.

He is currently an Assistant Professor with the University of São Paulo, São Carlos, Brazil, where he works in the areas of sensor systems, digital signal processing, and industrial automation. He is an Associate Editor of IEEE Transaction on Instrumentation and Measurement Journal.

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