Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

An Intelligent CNC Machine Tool Based on Random Forests Parameters Prediction Model

Karlo Griparic

kgriparic@unipu.hr

Sveuciliste Jurja Dobrile u Puli https://orcid.org/0000-0003-2396-7487

Erik Otovic

Sveuciliste Jurja Dobrile u Puli

Diego Susanj

Sveuciliste Jurja Dobrile u Puli

Research Article

Keywords: CNC machining, data-driven modeling, machining parameter extraction, process planning

Posted Date: August 24th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10629072/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

An Intelligent CNC Machine Tool Based on Random Forests Parameters Prediction Model

Karlo Griparić[1*], Erik Otović[1] and Diego Sušanj[1]

[1*]Faculty of Engineering, Juraj Dobrila University of Pula, Croatia.

*Corresponding author(s). E-mail(s): kgriparic@unipu.hr;

Abstract

Conventional computer numeric control machining production planning often depends on expert knowledge, leading to long preparation times and inconsistent parameter selection. This paper introduces a data-driven workflow for automated machining parameter optimization, utilizing insights from previously run G-code programs. A developed analyzer identifies four milling operations and their corresponding parameters, while a random forest regression model predicts milling parameters based on factors like operation type, workpiece material, and tool diameter. The study covered three machining centers with 251 operations analyzed. Results indicate that tool diameter, operation type and machine identity are key predictors of machining parameters. This approach allows G-code based production environments to leverage accumulated shop-floor knowledge, improving consistency in CNC setup and process planning. A two-stage automated pipeline analyzes new G-code programs, suggesting suitable machining parameters while allowing operator discretion over final decisions. The method requires no hardware modifications, and has been experimentally validated to enhance machining processes.

Keywords: CNC machining, data-driven modeling, machining parameter extraction, process planning

1 Introduction

Complex and customized products have placed significant pressure on manufacturing systems to become more agile and flexible [4, 29]. Automated manufacturing systems, which consist of multiple autonomous workstations, are a common approach in modern manufacturing processes. The relationships and interconnections between these workstations are essential for achieving the desired outcomes of the entire manufacturing system. In recent decades, global demands have required increases in flexibility, product diversity, and production speed, while at the same time product lifetimes have become very short.

Although computer controlled machine tools, or CNC (Computer Numerical Control) machines, have been widespread since the mid-20th century, transformations in production paradigms and growing product complexity have renewed interest in high-performance CNC machine tools. CNC machines, available in many different configurations, play an essential role in production due to their ability to process various types of materials quickly and accurately while ensuring product quality, repeatability, and flexibility. To attain optimal efficiency in producing specific mechanical components within modern manufacturing processes, these workflows must be fully digitalized [20]. This objective aligns with the emerging paradigm often referred as Industry 5.0. Effective toolpath generation is crucial in computer controlled machine tools, as it optimizes machining accuracy, surface finish quality, tool life, and

program execution time. The process of planning operations for manufacturing a machine part on CNC machine tools entails creating technological data that outlines the sequence of machining operations.

Setting up the production of machined parts using machine tools begins with an experienced engineer. The engineer must determine the required operations, assess the available machine tools and equipment, design the fixture, and write or generate the program in G-code [5]. Finally, after setting up the machine and tools, a prototype can be manufactured that meets the specified quality requirements. For instance, the parameters of the same or similar machining operation conducted on two different machines can vary significantly. The differences in processing time can be attributed to factors such as maximum cutting and rotational speed, which are limited by the machines' design [16]. Robust machines, for example, can deliver more power for machining. Furthermore, they are heavier, which often helps eliminate vibrations caused by cutting and reduces the tool's wear or the risk of breakage. Other factors that can influence cycle time include the duration of the automatic tool exchange, axis travel limits, spindle construction, and similar factors [35]. The central question studied in the paper is how to optimise the described setting process. In many manufacturing environments, there is an inherent trade-off between the need of production under the suboptimal and optimal processes. The mentioned traditional setup approach is classified as a suboptimal process because it depends on skilled personnel to configure the machine tool for production.

The aim of this study is to develop a data-driven model for extracting machining parameters while accounting for key technical factors, including the machine type and specifications, workpiece material, cutting tool performances, and machining strategies. Unlike prior program-mining work that induces rules for a single machine or part family [13], predicts a subset of parameters from richer attribute graphs [6], or captures knowledge without a predictive model [36], we train a single multi-output regressor that jointly predicts spindle speed, feed, and radial/axial depth of cut across three heterogeneous production machining centres, using only features recoverable from validated G-code. The transition from a manual, expert-dependent setup to an intelligent CNC system is achieved by integrating a random forest (RF) regression algorithm into the existing production workflow. To summarize, the main contributions of the paper are:

1. A modified production workflow that reduces CNC machine-tool setup time, supports machining quality across different platforms, and improves production planning through more accurate operational-time prediction.
2. A RF regression model generated from validated programs that predicts the most important machining parameters.
3. An automated pipeline for suggesting cutting parameters for new programs by recognizing machining operations and recommending working parameters according to the platform, workpiece material, and tool characteristics.

The remainder of this paper is organised as follows. Section 2 reviews related work on adaptive and intelligent CNC control, describes the limitations of G-code, presents alternative programming paradigms, and discusses the application of machine learning (ML) to machining processes. The proposed ML approach is described in Section 3, which introduces the machining features database, defines the model input and output parameters, and presents the developed parsing algorithm. Section 4 presents the RF algorithm adopted for multi-output regression and provides its mathematical formulation. The experimental setup, descriptive statistics of the training dataset, developed RF model, cross-validation methodology, and verification of the two-stage pipeline are presented in Section 5. Finally, Section 6 discusses the significance of the obtained results and provides concluding remarks.

2 Related work

A G-code is a set of commands used to control the movement of automated machine tools. It was developed alongside the automation of machine control and has become a fundamental programming method for a wide variety of machine tool configurations. Thus, it is obvious that G-code faces limitations in flexibility and adaptability when it comes to new planning and programming techniques. To overcome these limitations,

the literature presents two possible approaches. The first strategy involves upgrading current manufacturing processes to become adaptive and self-learning [17, 34]. This way, the machine controller requires minimal or no changes to implement necessary improvements toward a networked manufacturing industry [33]. The second approach utilises a novel principle that integrates the entire software solution into a single chain [18, 24, 37]. It replaces G-code and enables bidirectional data flow, allowing for real-time simulation, optimisation, and multi-process supervision across diverse manufacturing technologies. While this second approach provides greater opportunities for implementing artificial intelligence (AI) algorithms, it would require all controllers installed on machine tools to be replaced, which is a significant challenge. This retrofitting process is extremely expensive and often not affordable for small and medium enterprises (SMEs) [28].

On the other hand, the production workflow and machining operations can be improved by implementing available technologies, including optimisation principles, simulations, and AI. Recent work has proposed an optimisation method for a multi-spindle machining centre to reduce non-cutting time via coordinated spindle and tool change strategies [11]. The use of advanced sensors, networks, and computational technologies has led to the introduction of digital twins (DT). Recent work on designing a DT for a CNC machine tool has been presented in [22], where the authors proposed a new modeling framework segmented into four layers that eliminates uncertainties in the real manufacturing process. The verification of the proposed framework was conducted on a turning machine. Recent surveys in cognitive robotics review how AI and ML techniques are shaping advanced robotic systems and their control pipelines [27]. While the theoretical link between Industry 5.0 and sustainability is well-established, actual implementation is rare, especially among SMEs. The authors [10] propose a methodology that could bridge the gap between high-level strategic sustainability goals and the practical, operational realities of manufacturing through a structured twin transition. A recent survey [21] explores the integration of ML in CNC milling to enhance manufacturing precision and efficiency. It evaluates algorithms like neural networks, support vector machines, and deep learning. The study emphasizes data preprocessing and feature engineering for reliable predictive systems in smart factories, highlighting how digitalization minimizes production errors and prolongs cutting tool life through intelligent monitoring. Closely related to the approach adopted in this paper, [7] developed a data-driven parameter planning method for NC machining of structural parts, in which process parameters are derived from historical machining data rather than purely analytical cutting models, demonstrating the feasibility of data-driven parameter selection for computer-controlled machining.

Additional novel methodology applied in machining includes ML algorithms that are often used to predict tool's condition. Among classical methods, authors [8] introduced the use of artificial neural networks for determining the wear of cutting tools. A solution to a similar problem has been given in [31], where the authors investigated the ability to predict tool wear in milling operations using RF methods. The support vector machine method is developed in [19] for online evaluation of a cutting tool's degradation state. ML includes a range of methods and algorithms designed to derive insights from data and to enhance their performance over time by learning from past experiences [12]. ML is generally divided into three major categories: Supervised Learning (SL), Unsupervised Learning (UL), and Reinforcement Learning (RL) [1]. SL is a type of predictive learning in which algorithms learn to identify patterns and relationships from a labelled dataset. The main goal of a supervised model is to accurately approximate the relationship between an input vector and an output vector, enabling it to make reliable predictions on new, unseen data. SL can be divided into two main categories based on the nature of the output variable. The first category is classification, which predicts categorical output variables. Common industrial applications of classification include detecting process failures, identifying specific types of equipment faults, and classifying the quality levels of production batches. The second category is regression, which predicts continuous output variables. Examples of regression tasks include estimating physical properties such as material thickness and surface roughness, as well as predicting design metrics like power consumption and CPU performance.

A RF regressor for face milling that predicts material removal rate, surface roughness, and active energy consumption from cutting speed, feed rate, depth and width of cut are published in the study [2]. The authors [25] introduced a ML approach that evaluates G-code for CNC milling processes to estimate energy demand from basic G-commands, combining traditional machining processes with data driven techniques.

3 Methodology

A three-dimensional digital model of a part can be created using computer-aided design (CAD) software. The accompanying technical drawing provides additional information needed for manufacturing the part, including details about tolerances, surface roughness requirements, threads, and other important notes. Having both the digital model and the technical documentation is a prerequisite for creating a program for a numerically controlled machine tool. As mentioned earlier, the programming language used by these machine tools is known as G-code. The second step in the production workflow is to generate the paths that the tools should follow to shape the material into the desired geometrical forms. There are essentially two possible approaches to this process. The first involves programming through manual movements, predefined macros, and cycles. This type of programming can only be conducted for simple geometric operations, typically referred to as 2.5D programming. These operations include drilling, face milling, and operations that are following a straightforward path. This approach, while limited in path-generation options, is human-oriented and highly optimised due to its compact structure. Additionally, it facilitates an exceptionally easy modification process, which can save significant time in various practical onsite situations.

The second approach is to use computer aided manufacturing (CAM) software tools, which provide a partly automated way to generate instructions for a given machine. The main advantage of CAM software is its ability to create complex paths. However, experienced personnel are still required, as they should possess knowledge of both software and the physical machining processes, as well as the programming languages for various machine controllers. Even though CAM software can access appropriate information about tools and their machining capabilities, which is stored in an adequate database, faster and more reliable program generation still requires a lot of pre-production actions. Often, the adjustments made online can significantly differ from the programs generated by the postprocessor. Furthermore, because CAM software is generally a commercial application, there is limited information available on how to extract parameters from modified and tested programs for future G-code generation.

In this paper, we propose a novel and independent approach that utilizes ML algorithms to extract patterns and parameters from tested and generated programs. These algorithms generate a database with all relevant data about the machining process. Based on this data, a new automated parameter optimization algorithm identifies and recognizes all operations in the G-code and optimizes them according to recommendations generated from the model. To prevent potential mistakes in optimization, we have implemented a maximum change limit that the algorithms can make without the machinist's approval. The complete workflow for machine preparation, G-code generation, and ML based optimization is illustrated in Fig. 1.

3.1 Machining features database

G-code programming language in CNC machine tools defines spindle rotation, axis movements, feed rates, tool changes, and auxiliary operations during machining. In a typical program block, commands for spindle activation, feed definition, and coordinate positioning are represented by capital letters (S, F, X, Y, Z, M, etc.) followed by numerical values, usually without spaces or additional symbols. The core commands in this structured format are defined by ISO standards [14], with possible variations depending on the machine manufacturer. This structure enables deterministic machining behavior and supports portability across different machine tools. A comprehensive overview of operational principles and G-code commands can be found in [30]. A short example of a CNC program segment with explanation in the brackets is shown in Listing 1.

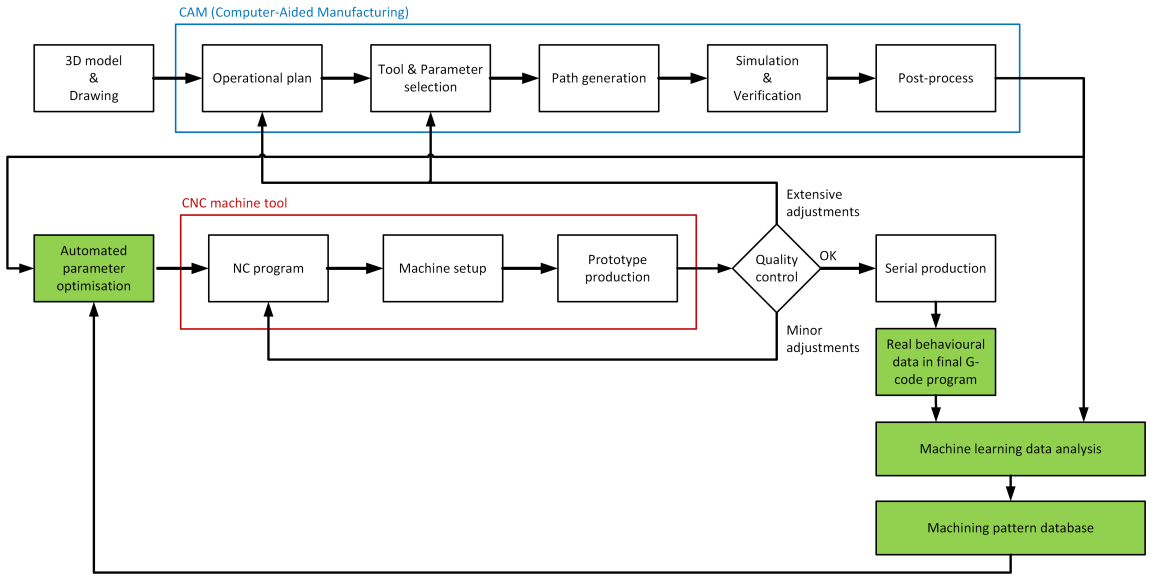


Fig. 1 The workflow outlines the operational steps involved in setting up the production of mechanical parts using numerically controlled machine tools. In addition to the traditional tasks (represented by white boxes) required to generate a G-code program, we have proposed a ML approach (represented by green boxes) to accelerate the parameter definition process.

```
OX1 OY1 OZ1 (ACTIVATE COORIDATES)
T1 M6 (SET TOOL)
S1500 M3 (SET ROTATIONAL VELOCITY)

G0 X0 Y0 Z50 (FAST MOTION)
G0 Z20
G1 Z-2 F300 (LINEAR MOTION)
G1 X50 F1000
G3 X50 Y40 I50 J20 (CIRCULAR MOTION)
G1 X0
G0 Z5

M5 (STOP ROTATION)
M2 (END OF PROGRAM)
```

Listing 1: Example G-code program segment (with explanatory comments in brackets).

In this work, we analyzed the machining capabilities of a milling machine and focused on several input parameters that significantly affect the material removal rate. The proposed model examines the four most commonly used conventional operations in the milling machining process. Additionally, we considered the production of items made from several different types of materials. Another important factor influencing the machining process is the technical specifications of the tools used. Finally, we identified the key machining parameters that can be adjusted to achieve the desired product quality, optimize operation time, and reduce tool wear, as well as other important production variables. These parameters include spindle rotational speed, feed rate, and the tool's cutting width and depth. The description and range of input and output parameters are shown in Table 1. Each input parameter is classified into several categories. In this work, we focused exclusively on the milling machining process and further specified machine characteristics by identifying the machine number M_i. For example, the machining parameters for producing the same item on two different machine tools may vary due to differences in the machines' construction specifications and output power. In other words, a machine with higher output power can perform certain milling operations faster due to its robust mechanical design. On the other hand, the output parameters are continuous and can be set to a minimum value required for the operation. Parameters such as spindle rotational speed, feed rate, cutting width, and depth can be calculated in advance using well-known traditional algebraic equations [26]. However, on-site conditions such as the type of operations, cutting conditions, machine specifications, and cooling abilities can significantly influence machining quality and production time.

Table 1 Description and values of machining parameters in CNC programs.

	Parameter	Value
INPUTS	Machine type	Mill
	Machine number	$[M_1, ..., M_i, ..., M_m]$
	Workpiece material	$[W_1, ..., W_i, ..., W_w]$
	Operation type	Drilling
		Face milling
		Contour milling
		Pocket milling
	Tool	Tool diameter
OUTPUTS	Spindle speed $\boldsymbol{n}$	0 – 20000 **[rpm]**
	Feed $\boldsymbol{f}$	0 – 15000 **[mm/min]**
	Cut width $\boldsymbol{a_w}$	0.01 – D_{max} **[mm]**
	Cut depth $\boldsymbol{a_d}$	0.01 – 20 **[mm]**

3.2 Conventional parameters determination

Traditionally, spindle speed and feed rate in machining are determined through analytical calculations based on tool geometry and material cutting data [15]. Specifically, the spindle speed n $[rpm]$ is calculated using the recommended cutting speed v_c $[m/min]$ and the tool diameter D $[mm]$:

$$n = \frac{1000\, V_c}{\pi D}. \tag{1}$$

This method provides a systematic starting point for setting machining parameters. The feed rate F $[mm/min]$ is then obtained by multiplying the feed per tooth f_z $[mm/tooth]$, the number of cutting teeth on the tool z, and the spindle speed n:

$$F = f_z\, z\, n. \tag{2}$$

3.3 Selection of a machine learning algorithm

Although many machine learning algorithms could handle this problem, we have chosen RF because it can be applied to a variety of data types. It can handle both classification tasks, where the response variable is categorical, and regression tasks, where the response variable is continuous [9]. In the generated machining features database, described in the previous section, the input feature set is heterogeneous, combining three categorical variables: machine identifier, workpiece material, and operation type, with six numeric variables: tool number, tool diameter, spindle rotational speed, axis travel velocity, cutting width and depth. RF can deal with problems where the number of features exceeds the number of samples. In our case study, we expect a small dataset for some machines, which is often the case in real facilities where machine tools can operate in different series. Hence, for larger series, we expect fewer available programs.

RF algorithms are widely recognised for their simplicity and robust performance across diverse applications, with predictions based on a decision-tree ensemble learning approach. This structure not only enhances predictive accuracy but also reduces the risk of overfitting [23]. Another important advantage is the minimal parameter tuning required to achieve optimal results, which accelerates model development. These combined strengths make RF a preferred choice in many real-world scenarios, particularly when working with heterogeneous datasets or limited data samples.

With these advantages in mind, and given the dataset characteristics described above, RF is an appropriate modelling choice for automated machining parameter prediction in CNC machine tool applications.

4 Random forest algorithm

This supervised ML approach is specifically designed to bridge the gap between theoretical CAM-generated programs and the real-world adjustments made by experienced operators. The

goal of RF is to create a ML model by utilising an ensemble of decision trees [3, 32, 38]. The model effectively maps complex relationships between input variable, such as machine type and specifications, workpiece materials, operation categories, and tool type, and the continuous output variables required for machining.

We consider a ML model composed of N decision trees. Each tree is constructed from N generated datasets acquired through resampling with replacement from the original training set. This process is known as bootstrapping, and its main advantage is that it increases the diversity of the training data. Let $D = \{D_1, ..., D_i, ..., D_N\}$ represent N random subsets of dataset, where each subset D_i is sampled independently of other subsets $D_1, ..., D_{i-1}$, and is defined as:

$$D_i = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$$

where X_i and Y_i are the input and output vectors, while n is the size of the dataset. In our model, the input variables are extracted from G-code programs successfully run in serial production, where each program typically consists of several machining operations. The input vector $X_i = (M_i, W_i, O_i, T_i)$ contains the following variables: M_i denotes the unique machine identifier used in the observed workshop, W_i denotes the workpiece material (classified into several categories), O_i denotes the milling operation type, and T_i denotes the tool diameter. The corresponding output vector $Y_i = (N_i, F_i, A_{w,i}, A_{d,i})$ comprises the spindle rotational speed, feed rate, width of cut, and depth of cut, respectively.

It should be noted that in each subset D_i, some (X_i, Y_i) pairs may be repeated as sampling is done with replacement. The number of regression trees represents the complexity of the model and is the hyperparameter defined by the user. Therefore, the training process begins by bootstrapping datasets.

The next step in building a ML model is the creation of the decision trees, one for each D_i subset. The main question is how to split the input variables in the tree. The splitting criterion used during the construction of a decision tree plays an important role in the efficiency of the developed RF model. The goal is to reduce the variance of the generated trees on training data. Each regression tree begins at the root node and splits into two child nodes. The splitting process is recursively performed for newly created nodes until reaching the leaf node, which is defined by a chosen stopping condition. At each node, the algorithm randomly samples m variables from the available input set X_i and uses these variables as candidates for splitting at each non-leaf node. The described methodology, called bagging, reduces correlation among generated trees. The tree growing process is an iterative procedure that continually evaluates and splits the data based on the chosen measure, seeking to create a model that effectively categorises or predicts outcomes with adequate accuracy. Thus, the nodes' splitting criteria are basically an optimisation problem. The main goal in the creation of a decision tree is to find the minimal impurity of the entire dataset at each node. This impurity can be quantified using different metrics, depending on the type of task at hand. For classification tasks, measures like information entropy or Gini impurity are commonly employed. Information entropy gauges the uncertainty in the data, while Gini impurity assesses the likelihood of misclassifying a randomly chosen element from the dataset. Both methods aim to ensure that the resultant nodes contain samples that are as homogeneous as possible. In contrast, for solving a regression problem, the focus is on reducing the residual sum of squares, which calculates the total deviation of the predicted values from the actual values. By minimising this residual, the regression decision tree seeks to achieve better predictions, ensuring that the output is as close as possible to the true values.

Suppose a training dataset D_i for the i^{th} decision tree is divided into K regions $R_1, R_2, \ldots, R_K$. The output value of the j decision tree is given with the following sum:

$$h_j(x) = \sum_{k=1}^{K} c_k I(x \in R_k), \tag{3}$$

where c_k is the mean target value in region R_k, and I is the indicator function. The first step in minimising the distance between splitting nodes is to determine the average of the output data y_i in each region R_k:

$$c_k = \frac{1}{n} \sum_{x_i \in R_k} y_i. \tag{4}$$

When a node in a decision tree is split, then we choose a splitting variable (v) and a split point (p). This creates a pair of half-planes, denoted as R_1 and R_2: $R_1(v,p) = \{X \mid X_v \leq p\}$: The first child node contains all data points where the chosen variable is less than or equal to the threshold. The second child node $R_2(v,p) = \{X \mid X_v > p\}$ contains all data points where the variable is greater than the threshold. The algorithm selects the specific half-planes that minimize the sum of squares for the target values (y_i) within those regions. It solves the following minimization problem to find the optimal v and p:

$$\min_{v,p} \left[\min_{c_1} \sum_{x_i \in R_1} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2} (y_i - c_2)^2 \right], \tag{5}$$

where c_1 and c_2 are the predicted constants for each half-plane given by Eq. (4). To summarize, the training dataset is classified into two regions. This process is iteratively applied to each resulting region until a stopping condition is met. The tree's maximum depth is defined through the stopping condition.

Finally, the relationship between the input vector $\mathbf{x}$ and the predicted machining parameters $\hat{\mathbf{y}}$ is represented by the averaging the predictions of N decision trees:

$$\hat{\mathbf{y}} = \frac{1}{N} \sum_{i=1}^{N} h_i(\mathbf{x}). \tag{6}$$

The overall structure of the proposed RF prediction model is illustrated in Fig. 2.

Three crucial parameters in the RF algorithm significantly influence the effectiveness and accuracy of the ML prediction model. Firstly, the user should define the total number of regression trees in the forest ensemble. It governs ensemble size and prediction stability. Then, during tree construction, a stopping criterion plays an important role, because it defines the tree's depth and influences the high variance of the model. The third important parameter to define is the number of randomly selected variables. It reduces the correlation between trees.

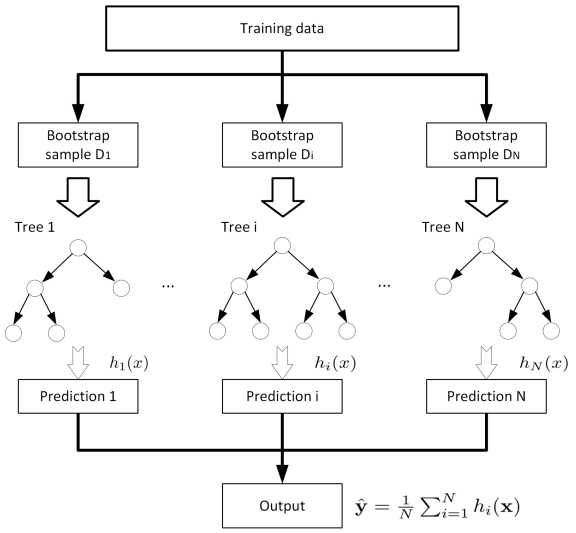


Fig. 2 A sketch of the machining parameters prediction model based on RF decision trees constructed from the training dataset.

5 Experimental verification

In this section, we have experimentally evaluated the effectiveness of the proposed RF algorithm for predicting machining parameters across three distinctly different milling machine centers. Table 2 summarizes the technical characteristics of the machining centers used in our study.

5.1 Experimental setup

The control unit of a CNC machine tool interprets programs written in G-code and primarily controls the movement of the machine's axes necessary for shaping a specific part. However, G-code commands may vary depending on the preferences of the control unit's manufacturer. In this paper, we have used a *D.Electron Z32* control unit model installed on all machines. The dataset used in this study was assembled by automatically parsing a collection of CNC part programs. A Python parser was developed to extract machining parameters directly from the program files, including header metadata, operation type recognition, tool identification, the specified spindle rotational speed, parametric variable assignments, recognized feed values, and subroutine blocks reached via jump commands.

Table 2 Technical specifications of the CNC machining centres used in the study.

Characteristic	BIESSE Rover 336	SCM Routronic	LIDWEL
Number of axes	3	3	3
Travel distances (X,Y,Z) [mm]	(3 360, 1 320, 150)	(3 200, 1 400, 400)	(800, 400, 350)
Max. Spindle speed [rpm]	20 000	18 000	2 500
Spindle power [kW]	7.5	7.5	15
Feed rate range [mm/min]	0–15 000	0–15 000	0–5 000
Tool holder	ISO 30	ISO 30 and ISO 40	ISO 40

5.2 Training data

After parsing and filtering, the dataset comprises 251 machining operations. Three types of workpiece materials were identified. Table 3 summarises the composition of the dataset by machine platform, workpiece material, and operation type. The four milling operations identified represent fundamental machining techniques. However, they cover only a portion of the possible machining operations. Since these operations are among the most widespread, we have chosen them to verify the proposed contributions experimentally. As previously noted, the machining process is influenced by numerous factors that affect both its efficiency and the quality of the finished product. Although machining processes can be repeated in series of varying sizes, in the most cases where medium or large series are produced, each machine has a limited number of verified operations. This complexity and data availability limitation pose a significant challenge for many ML algorithms in generation of a prediction model.

Table 4 presents the mean and standard deviation of the four machining parameters: spindle speed n [rpm], feed rate f [mm/min], cut width a_w [mm], and depth of cut a_d [mm]. Substantial differences in operating regimes are evident between machine tools. The dataset obtained only programs where the workpiece material is an engineering plastic.

Spindle rotational speed setups recorded as zero (arising from programs where the spindle command is issued in a preceding subroutine block) were treated as missing values for statistical purposes, yielding an effective spindle speed missing rate of 2.0 %. The feed rate missing rate was 0 %.

Table 3 Composition of the machining operation dataset ($N = 251$).

Category	Subcategory	Count
Machine centre	LIDWEL	101
	SCM	94
	BIESSE	56
Workpiece material	Polypropylene (PP)	208
	Polyethylene PE500	31
	Polyethylene PE1000	12
Operation type	Contour milling	88
	Drilling	65
	Pocket milling	64
	Face milling	34

Fig. 3 and 4 illustrate the distribution of programmed spindle speeds and feed rates, respectively, for polypropylene (PP) workpieces, categorized by operation type across different machine centers. Contour milling shows the widest variation in spindle speeds among the machines. The SCM primarily operates within the speed range of 2000 to 5000 rpm, with most programs clustered near the observed ceiling of 5000 rpm. In contrast, the BIESSE machine operates over a broader range of 1300 to 3800 rpm, while the LIDWEL is confined to a narrow band of 1700 to 1900 rpm. These differences reflect the varying spindle capabilities of the machines and the distinct tool diameters used. For drilling, all three machines exhibit a greater spread in speeds compared to their contour milling distributions. The SCM reaches the highest spindle speeds, ranging from 900 to 5000 rpm, while the LIDWEL is limited to between 1000 and 1800 rpm. Face milling is performed on all three machines. On the BIESSE, the speeds range from 1900 to 3500

Table 4 Descriptive statistics of machining parameters by machine centre (mean ± standard deviation).

Machine	n [rpm]	f [mm/min]	a_w [mm]	a_d [mm]
BIESSE	2436 ± 832	2175 ± 997	1.5 ± 2.2	19.5 ± 20.8
SCM	3026 ± 1488	1603 ± 822	2.4 ± 2.1	15.1 ± 15.1
LIDWEL	1732 ± 245	2112 ± 658	1.9 ± 2.3	20.4 ± 20.0

rpm, concentrated around 3000 rpm. The LIDWEL operates between 1700 and 1900 rpm, while the two recorded SCM programs run at 4000 and 5000 rpm. This distribution is consistent with the use of large-diameter face mills, where even moderate spindle speeds can achieve adequate peripheral cutting velocities. Pocket milling data is conducted on all three machines. The SCM again shows the widest spread, from 1500 to 5000 rpm, the LIDWEL machine is concentrated between 1800 and 2000 rpm, while the BIESSE has only two recorded programs, both at 2800 rpm. Across all operation types, the LIDWEL consistently records the narrowest and most conservative spindle-speed bands, consistent with its comparatively lower rated maximum spindle speed of 2500 rpm.

Face milling exhibits the highest and most consistent feed rates, which range from 2000 mm/min to 4500 mm/min, with a median of 2750 mm/min. This reflects the use of large diameter face mills, allowing for high table feeds while maintaining moderate chip loads per insert. In contrast, contour milling and pocket milling show much wider feed rate distributions, spanning from 600 mm/min to 5000 mm/min and from 600 mm/min to 4000 mm/min, respectively, while drilling spans the most conservative range, from 250 mm/min to 2500 mm/min. This variability is consistent with the diverse range of tool diameters used in these operations. From small diameter end mills and drills that require conservative feeds to ensure dimensional accuracy, to larger cutters that can sustain high material removal rates. Overall, the trend across all operations demonstrates that face milling maintains the highest feed rates, followed by contour and pocket milling, while drilling utilizes the most conservative feeds. This is consistent with the need to limit thrust force and preserve hole quality during full-depth drilling passes in engineering thermoplastics.

The comparison between the programmed spindle speed against the theoretical value predicted by a constant cutting speed relation ($V_c = 100\,\mathrm{m/min}$) for contour milling operations on PP across all three machines is given in Fig. 5. For small tool diameters (6 mm to 16 mm), the SCM records track the theoretical curve most closely, with an actual to theoretical ratio between 0.94 and 1.35, whereas BIESSE scatters both above and below the curve and LIDWEL consistently falls below it. Beyond a diameter of approximately 20 mm, the actual spindle speed increasingly departs from the theoretical prediction. The speed values of BIESSE exceed the theoretical curve by a factor of up to 3.5 at the largest recorded diameter (32 mm). The LIDWEL, despite operating predominantly at large diameters, maintains an almost constant spindle speed of 1800 rpm to 1900 rpm across the entire 6 mm to 44 mm range instead of following the expected inverse relationship with tool diameter. This indicates that, for larger tools, spindle speed selection is governed less by a constant surface cutting-speed strategy and more by practical constraints such as maximum spindle speed limits, tool manufacturer recommendations, or operator habit. The deviations in programmed spindle speed from the theoretical curve illustrate the limitations of a single constant cutting speed model and further motivate the development of the data-driven RF approach adopted in this work.

Fig. 6 represents the distribution of programmed feed rates for contour milling operations performed on the BIESSE machining centre, disaggregated by operator. Three operators contributed to the dataset: OP1 ($n = 5$), OP2 ($n = 10$), and OP3 ($n = 12$). Despite working on the same machine platform and operation type, the three operators exhibited different feed rate strategies. OP1 programmed the highest and most dispersed feed rates, ranging from 1500 mm/min to 5000 mm/min with a

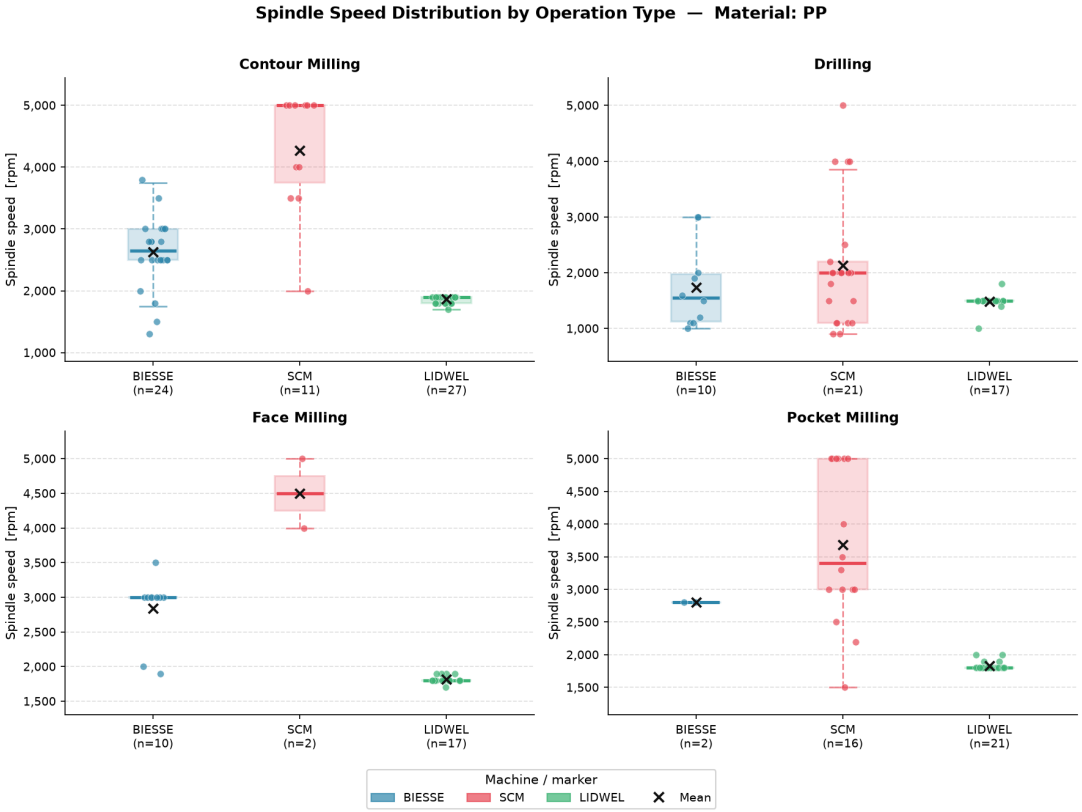


Fig. 3 Spindle speed distribution by machining operation for plastic (PP) workpieces.

mean of 2860 mm/min, indicative of an aggressive cutting approach, although the small sample size limits the strength of this conclusion. OP2 programmed feed rates predominantly in the range 700 mm/min to 3300 mm/min, with a median of 2100 mm/min, reflecting a moderate and consistent cutting strategy. OP3, who contributed the largest number of records, programmed the most tightly clustered feed rates, with a range of 1800 mm/min to 2525 mm/min, a median of 2000 mm/min, and overall values spanning 600 mm/min to 3000 mm/min.

These persistent differences in operator strategies, even after accounting for machine platform and operation type, indicate that feed rate selection at the studied facility is largely influenced by individual experience and personal preferences rather than a standardized process plan. This variability directly affects key production metrics, including part cycle time, tool life, and surface quality consistency. Consequently, operators' decisions introduce significant bias into the dataset, which can compromise the reliability of traditional rule-based models. These observations highlight the practical need for a data-driven parameter prediction model that can recommend optimal spindle speed and feed rates independent of operator tendencies, thereby improving consistency and objectivity in machining operations.

5.3 RF model validation

Four input features were derived from the parsed G-code records: three categorical variables (machine identifier, workpiece material, operation type) encoded with label encoding, and one numeric variable (tool diameter in mm). The tool number was excluded from the feature set, as it is an arbitrary program-specific identifier that carries no physical information about the machining process. The four machining parameters (n, f, a_w, a_d) are predicted by RF estimator. The number of regression trees is an empirical value that should be chosen to optimise model complexity and prediction accuracy. The literature commonly recommends setting the number of trees between

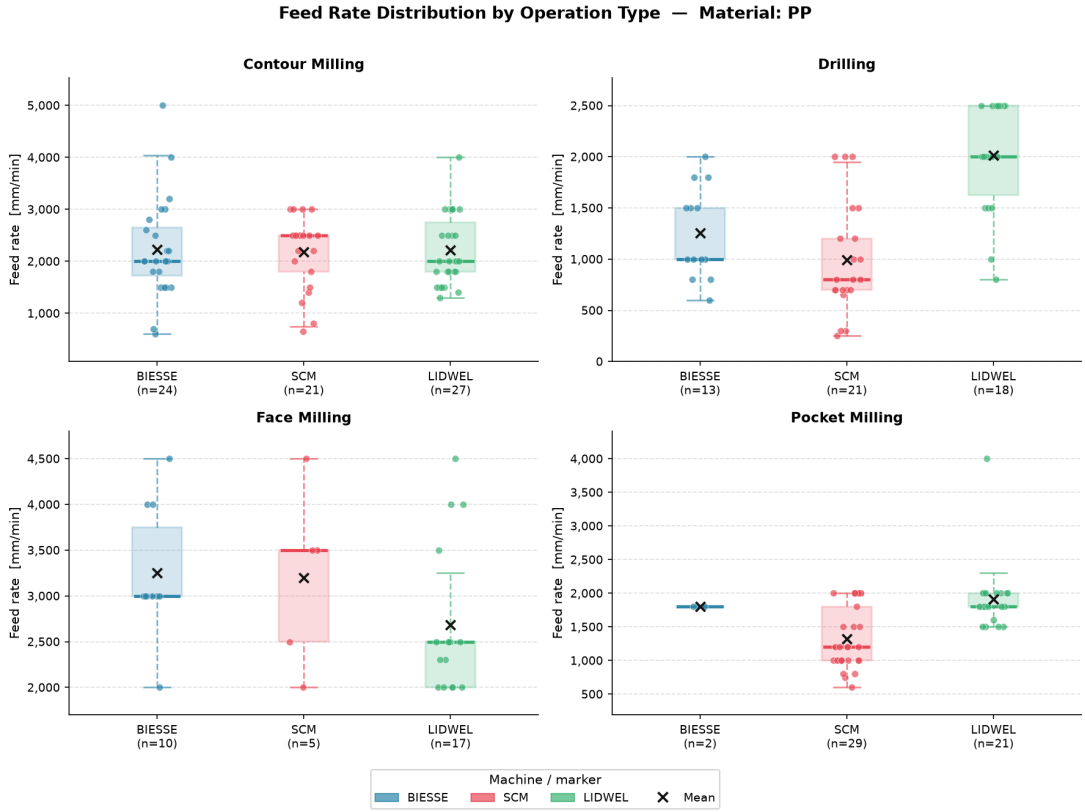


Fig. 4 Feed distribution by machining operation for plastic (PP) workpieces.

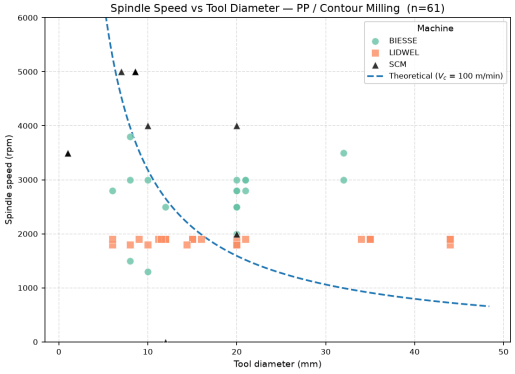


Fig. 5 The actual spindle speed data compared with the theoretical value assuming a cutting speed of 100 m/min.

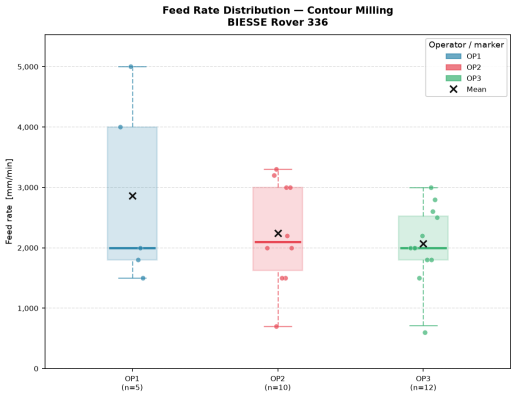


Fig. 6 Feed-rate distribution by operator for BIESSE machining programs.

100 and 500, as this range typically yields reliable performance. In our case, we found that using 200 trees achieves satisfactory predictive results.

The performance of the trained RF regressors was evaluated using k-fold cross-validation, a resampling procedure designed to provide an unbiased estimate of generalisation error when the available dataset is too small for a dedicated test set. The dataset comprised 251 records, of which 1 record (0.4 %) was dropped due to a missing value in at least one of the four target parameters, yielding 250 valid records used for model training and cross-validation. In the cross-validation process, the dataset is partitioned into k mutually exclusive subsets of approximately equal size. Since the four

target variables are continuous, folds are formed after a random shuffle of the records. In our study, a 5-fold cross-validation is employed to evaluate the model. In each of the five iterations, the model was trained on the union of 4 folds (approximately 200 records) and evaluated on the remaining fold (approximately 50 records). The evaluation metric was the coefficient of determination R^2, defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{7}$$

where y_i is the observed value, $\hat{y}$ is the model prediction, and $\bar{y}$ is the mean of observed values in the test fold.

Table 5 reports the 5-fold cross-validated R^2 obtained for each of the four predicted machining parameters, together with its standard deviation across folds. Spindle speed n achieves high prediction accuracy, with $R^2 = 0.751 \pm 0.062$, ranging from 900 rpm to 6000 rpm across the dataset. This accuracy is primarily influenced by machine identity together with tool diameter, as detailed in Table 6, which highlights the three most significant features in the prediction model. This finding aligns with the observation that each machine center tends to operate within a specific and consistent spindle-speed range, as also noted in Table 4. The cut width a_w exhibits near-perfect predictive accuracy, with $R^2 = 0.992 \pm 0.009$. This high accuracy is expected due to its close mechanical relationship with tool diameter, which is the dominant feature identified for this parameter. Conversely, the cut depth a_d has the lowest prediction score, with $R^2 = 0.363 \pm 0.128$. This outcome is expected because the initial cutting depth is often unknown due to a lack of information regarding the material removal rate at the start of the operation. For instance, in face milling, the removal rate during the first pass can only be estimated. The developed parser does not have this information and consequently filled the data with zeros. This has a significant impact on the accuracy of the trained RF model.

Feature importance in a RF estimator was quantified using the mean decrease in impurity (MDI), also known as the Gini importance. For each decision tree in the ensemble, the reduction in node impurity, measured by the mean squared error for regression tasks, is accumulated and averaged across all 200 trees. Features that are selected earlier and more frequently during the tree-building process receive higher importance scores if they produce the largest reductions in prediction error. The MDI values are normalised to sum to unity across all features, enabling direct comparison of relative contributions. In the conducted experiments, tool diameter was the dominant predictor for three of the four target variables (feed rate, cut width, and cut depth), reflecting the close mechanical coupling between tool geometry and these parameters. Machine identity was the dominant predictor for spindle speed, capturing the specific mechanical and operating capabilities of each machine tool. Operation type consistently ranked among the top three predictors for all four targets, while workpiece material contributed only marginally to spindle speed and feed rate, ranking third for cut width and cut depth instead. Table 6 describes the three most important features determined by mean decrease in impurity (MDI).

Table 5 Cross-validated R^2 for each predicted machining parameter (mean ± standard deviation across 5 folds).

Target parameter	**R^2 (5-fold CV)**
Spindle speed n	0.751 ± 0.062
Feed rate f	0.410 ± 0.114
Cut width a_w	0.992 ± 0.009
Cut depth a_d	0.363 ± 0.128

6 Conclusion

The potential of the manufacturing process equipped with CNC machine tools can be fully exploited only if machining operations are adequately optimised. This paper addresses the practical challenge of improving CNC setup and programming efficiency through a novel, data-driven workflow based on ML principles. The traditional approach typically determines cutting parameters based on the chosen tool's shape and cutting characteristics for specific workpiece materials. However, data analysis shows that factors such as machine rigidity, spindle capability, the specific product portfolios assigned to each machine, and operator experience also play important roles in selecting the appropriate operating conditions.

Table 6 The most important features per predicted target parameter.

Target parameter	Top-3 features (ranked by MDI)
Spindle speed n	machine, tool diameter, operation type
Feed rate f	tool diameter, machine, operation type
Cut width a_w	tool diameter, operation type, material
Cut depth a_d	tool diameter, operation type, material

We introduced a two-stage pipeline predictor that extends conventional CAD/CAM practice by estimating machining parameters, such as the tool's rotational speed, axis feed rates, and cutting parameters, for new machining programs. A RF multi-output regression model was trained to simultaneously predict spindle speed, feed rate, cut width, and depth of cut. Five-fold cross-validation confirmed that tool diameter is the dominant predictor for three of the four target parameters (feed rate, cut width, and cut depth), while machine identity is the dominant predictor for spindle speed, with operation type consistently ranking among the top three predictors for all four parameters. Finally, the trained model was integrated into an automated machining parameter estimator that parses a new G-code program and extracts the context for each detected machining operation. Based on the extracted parameters, the RF model is applied to predict the four target parameters, which are then presented to the operator as suggested replacements alongside the original values. The operator retains full authority over the final program, in keeping with the human-centred philosophy of the modern industry approach.

Future work will focus on expanding the dataset to include additional machine tool configurations, enlarging the range of machining operation types, validating optimisation performance across a broader range of part families, and integrating closed loop feedback from shop floor measurements.

Acknowledgements. This work was funded by the EU NextGeneration under the Juraj Dobrila University of Pula institutional research project number IIP UNIPU 010153-581.

Declarations

- **Funding:** This work was funded by the EU NextGeneration under the Juraj Dobrila University of Pula institutional research project number IIP UNIPU 010153-581.
- **Conflict of interest:** The authors declare that they have no conflict of interest.
- **Ethics approval and consent to participate:** Not applicable.
- **Consent for publication:** Not applicable.
- **Data availability:** The datasets generated and analysed during the current study are available from the corresponding author on reasonable request.
- **Materials availability:** Not applicable.
- **Code availability:** The parsing and model-training code used in this study is available from the corresponding author on reasonable request.
- **Author contribution:** K.G. conceived the study, developed the G-code parser and RF model, and drafted the manuscript. E.O. and D.S. contributed to the methodology, data analysis, and manuscript revision. All authors read and approved the final manuscript.

References

[1] Bertolini M, Mezzogori D, Neroni M, et al (2021) Machine learning for industrial applications: A comprehensive literature review. Expert Systems with Applications 175:114820. https://doi.org/10.1016/j.eswa.2021.114820, URL http://dx.doi.org/10.1016/j.eswa.2021.114820

[2] Bhattacharya S, Chakraborty S (2023) Prediction of responses in a CNC milling operation using random forest regressor. Facta Universitatis, Series: Mechanical Engineering 21(4):685–700. https://doi.org/10.22190/FUME210728071B, URL https://doi.org/10.

22190/FUME210728071B

[3] Breiman L (2001) Random forests. Machine Learning 45(1):5–32. https://doi.org/10.1023/a:1010933404324, URL https://doi.org/10.1023/A:1010933404324

[4] Browning TR (2001) Applying the design structure matrix to system decomposition and integration problems: a review and new directions. IEEE Transactions on Engineering Management 48(3):292–306

[5] Davim JP (ed) (2008) Machining: Fundamentals and Recent Advances, 1st edn. Springer, London, https://doi.org/10.1007/978-1-84800-213-5, URL https://doi.org/10.1007/978-1-84800-213-5

[6] Deng T, Li Y, Chen J, et al (2021) Informed machine learning-based machining parameter planning for aircraft structural parts. The International Journal of Advanced Manufacturing Technology 117(11-12):3563–3575. https://doi.org/10.1007/s00170-021-07861-2, URL http://dx.doi.org/10.1007/s00170-021-07861-2

[7] Deng T, Li Y, Liu X, et al (2021) A data-driven parameter planning method for structural parts nc machining. Robotics and Computer-Integrated Manufacturing 68:102080. https://doi.org/10.1016/j.rcim.2020.102080, URL http://dx.doi.org/10.1016/j.rcim.2020.102080

[8] Fu P, Li J, Li W, et al (2010) Tool wear states recognition based on integrated neural networks. In: 2010 Sixth International Conference on Natural Computation. IEEE, p 1344–1348, https://doi.org/10.1109/icnc.2010.5583748, URL http://dx.doi.org/10.1109/ICNC.2010.5583748

[9] Genuer R, Poggi JM (2020) Random Forests with R. Springer International Publishing, https://doi.org/10.1007/978-3-030-56485-8, URL http://dx.doi.org/10.1007/978-3-030-56485-8

[10] Ghobakhloo M, Iranmanesh M, Foroughi B, et al (2023) Industry 5.0 implications for inclusive sustainable manufacturing: An evidence-knowledge-based strategic roadmap. Journal of Cleaner Production 417:138023. https://doi.org/10.1016/j.jclepro.2023.138023, URL http://dx.doi.org/10.1016/j.jclepro.2023.138023

[11] Griparic K, Susanj D, Matosevic V (2025) A control framework for the machining center with multiple spindles. Elektrotehni
vski Vestnik

[12] Huyen C (2022) Designing machine learning systems. O'Reilly Media, Sebastopol, CA

[13] Igari S, Tanaka F, Onosato M (2013) Customization of a micro process planning system for an actual machine tool based on updating a machining database and generating a database-oriented planning algorithm. Transactions of the Institute of Systems, Control and Information Engineers 26(3):87–94. https://doi.org/10.5687/iscie.26.87, URL http://dx.doi.org/10.5687/iscie.26.87

[14] International Organization for Standardization (2009) ISO 6983-1:2009 — automation systems and integration — numerical control of machines — part 1: Program format and definitions of address words. International Standard ISO 6983-1:2009, International Organization for Standardization, Geneva, Switzerland, URL https://www.iso.org/standard/34608.html

[15] Kalpakjian S, Schmid S (2022) Manufacturing processes for engineering materials in SI units, 6th edn. Pearson Education, London, England

[16] Kim K, Lee D (2023) Monitoring of machining process of machine tool based on artificial neural network using serial data. In: 2023 IEEE Smart World Congress (SWC). IEEE, p 639–643, https://doi.org/10.1109/swc57546.2023.10448651, URL http://dx.doi.org/10.1109/SWC57546.2023.10448651

[17] Kim S, Heo E, Lee H, et al (2024) Advanced adaptive feed control for cnc machining. Robotics and Computer-Integrated Manufacturing 85:102621. https://doi.org/10.1016/j.rcim.2023.102621, URL http://dx.doi.org/10.1016/j.rcim.2023.102621

[18] Latif K, Adam A, Yusof Y, et al (2021) A review of g code, step, step-nc, and open architecture control technologies based embedded cnc systems. The International Journal of Advanced Manufacturing Technology 114(9–10):2549–2566. https://doi.org/10.1007/s00170-021-06741-z, URL http://dx.doi.org/10.1007/s00170-021-06741-z

[19] Li Y, Huang X, Tang J, et al (2023) A steps-ahead tool wear prediction method based on support vector regression and particle filtering. Measurement 218:113237. https://doi.org/10.1016/j.measurement.2023.113237, URL http://dx.doi.org/10.1016/j.measurement.2023.113237

[20] Liu C, Xu X (2017) Cyber-physical machine tool – the era of machine tool 4.0. Procedia CIRP 63:70–75. https://doi.org/10.1016/j.procir.2017.03.078, URL http://dx.doi.org/10.1016/j.procir.2017.03.078

[21] Pasandidehpoor M, Nogueira AR, Mendes-Moreira J, et al (2025) Survey on machine learning applied to cnc milling processes. Advances in Manufacturing 13(4):847–885. https://doi.org/10.1007/s40436-025-00564-x, URL http://dx.doi.org/10.1007/s40436-025-00564-x

[22] Pei F, Chen S, Li Q, et al (2025) Construction method of cnc machine tool digital twin model based on the four-layer framework. IEEE Access 13:70211–70227. https://doi.org/10.1109/access.2025.3560647, URL http://dx.doi.org/10.1109/ACCESS.2025.3560647

[23] Ramesh K, Indrajith MN, Prasanna YS, et al (2025) Comparison and assessment of machine learning approaches in manufacturing applications. Industrial Artificial Intelligence 3(1). https://doi.org/10.1007/s44244-025-00023-3, URL http://dx.doi.org/10.1007/s44244-025-00023-3

[24] Rauch M, Laguionie R, Hascoet JY, et al (2012) An advanced step-nc controller for intelligent machining processes. Robotics and Computer-Integrated Manufacturing 28(3):375–384. https://doi.org/10.1016/j.rcim.2011.11.001, URL http://dx.doi.org/10.1016/j.rcim.2011.11.001

[25] Schmitt AM, Miller E, Engelmann B, et al (2024) G-code evaluation in CNC milling to predict energy consumption through machine learning. Advances in Industrial and Manufacturing Engineering 8:100140. https://doi.org/10.1016/j.aime.2024.100140, URL https://doi.org/10.1016/j.aime.2024.100140

[26] Shaw MC (2005) Metal Cutting Principles, 2nd edn. Oxford University Press, New York

[27] Soori M, et al (2023) Artificial intelligence and machine learning in cognitive robotics: A comprehensive review. Computers and Graphics https://doi.org/10.1016/j.cogr.2023.04.001

[28] Spaltini M, Terzi S, Taisch M (2024) Development and implementation of a roadmapping methodology to foster twin transition at manufacturing plant level. Computers in Industry 154:104025. https://doi.org/10.1016/j.compind.2023.104025, URL http://dx.doi.org/10.1016/j.compind.2023.104025

[29] Suh NP (1990) The Principles of Design. Oxford University Press, New York, NY, USA

[30] Suh SH, Kang SK, Chung DH, et al (2008) Theory and design of CNC systems, 2008th edn. Springer Series in Advanced Manufacturing, Springer, London, England

[31] Wu D, Jennings C, Terpenny J, et al (2017) Data-driven prognostics using random forests: Prediction of tool wear. In: Volume 3: Manufacturing Equipment and Systems. American Society of Mechanical Engineers, MSEC2017, https://doi.org/10.1115/msec2017-2679, URL http://dx.doi.org/10.

1115/MSEC2017-2679

[32] Wu D, Jennings C, Terpenny J, et al (2017) A comparative study on machine learning algorithms for smart manufacturing: Tool wear prediction using random forests. Journal of Manufacturing Science and Engineering 139(7). https://doi.org/10.1115/1.4036350, URL http://dx.doi.org/10.1115/1.4036350

[33] Wu Y, Yue N, Qian K (2022) Performance optimization of cnc machine tool system based on sensor data. Scientific Programming 2022:1–10. https://doi.org/10.1155/2022/5663824, URL http://dx.doi.org/10.1155/2022/5663824

[34] Xu X (2017) Machine tool 4.0 for the new era of manufacturing. The International Journal of Advanced Manufacturing Technology 92(5–8):1893–1900. https://doi.org/10.1007/s00170-017-0300-7, URL http://dx.doi.org/10.1007/s00170-017-0300-7

[35] Yao Y, Li Y, Chen Q, et al (2010) Optimize cnc milling parameters on-line. In: 2010 International Conference on Measuring Technology and Mechatronics Automation. IEEE, p 856–859, https://doi.org/10.1109/icmtma.2010.79, URL http://dx.doi.org/10.1109/ICMTMA.2010.79

[36] Yusuke UCHIDA Y, AOYAMA H (2007) Development of basic system to construct database of machining know-how. Proceedings of International Conference on Leading Edge Manufacturing in 21st century : LEM21 2007.4(0):7F605. https://doi.org/10.1299/jsmelem.2007.4.7f605, URL http://dx.doi.org/10.1299/jsmelem.2007.4.7F605

[37] Zhang Y, Yu X, Sun J, et al (2022) Intelligent step-nc-compliant setup planning method. Journal of Manufacturing Systems 62:62–75. https://doi.org/10.1016/j.jmsy.2021.11.002, URL http://dx.doi.org/10.1016/j.jmsy.2021.11.002

[38] Zhu M, Yang Y, Feng X, et al (2022) Robust modeling method for thermal error of cnc machine tools based on random forest algorithm. Journal of Intelligent Manufacturing 34(4):2013–2026. https://doi.org/10.1007/s10845-021-01894-w, URL http://dx.doi.org/10.1007/s10845-021-01894-w