POLITECNICO DI TORINO

Department of Management and Production Engineering

DEVELOPMENT OF A PREDICTIVE MAINTENANCE MODEL BASED ON COMBINATION OF PHYSICS-BASED AND DATA-DRIVEN METHODS

Supervisor
Prof. Bruno Giulia
Prof. Franco Lombardi
Dott. Emiliano Traini

Candidate
Gayratkhuja Akhrarov

Academic year 2021 – 2022

ACKNOWLEDGEMENTS

At the end of this long journey, I would like to spend a few words for all those who have always been there, without hesitation, both in times of difficulty and in happy and carefree ones.
First of all, a heartfelt thanks to my supervisor prof. Bruno Giulia, has been an perfect mentor, and thesis supervisor, offering her availability to discuss in all situtaions and for the precious advice she gave me throughout the induction and continuation of this work.
Also I would like to thank people without whom I would not have able to complete this research and made it through my masters degree.
And my biggest thanks to my family for all the support you have shown me through this journey and for believing in me during all my years of study.

ABSTRACT

Metal processing industry (milling, drilling, lathe) is the crucial industry for any industrial sectors, whether it is prototyping, production, construction, electronics. In metal processing industry processing of metallic parts mostly done by industrial tools(instruments) with certain specifications. There are several varieties of instruments depending on the purpose. As the only point of contact instruments in the process of machining, to the functionality of tool influence many factors, which is vibration, high temperature, swift temperature change, high revolving speed, melting of the processing part and even stability of machine and steadiness of part. All these factors influence to tool life and may cause irreparable and spontaneous wreckages. Considering these conditions, it is hard even for experienced technicians or engineers to predict exact moment for tool maintenance. The main issue is the early tool change or maintenance leads to ineffective use of resources and increasing the price of the product late change may lead, as stated above to even worse consequences (destroying the tool, part and may even machine) However, continuing industrialization and the high demands for mass production require the maximization of resources allocation and predictability of processes.

The innovations occur in intersection of different sectors and to solve the issue of tool unpredictable lifetime was proposed to use machine learning algorithms to predict the maintenance time of the tool. Machine learning algorithms is the emerging technology which using big amounts of data calculates and predicts the behaviors of the system based on existing data. For this research NASA-PCoE used several sensors of sound, vibration, temperature and created dataset to further development of techniques for creating efficient predictive models.

Nowadays, there are several libraries(toolkits) for developing machine learning predictive models and one of the most popular is Scikit based on Python. The purpose of this research is to develop several data-driven models for behavior prediction (Neural networks (Multi-layer Perceptron regressor) and compare them with the current techniques of physics-based methods (Taylor).

The purpose of this thesis is to develop Python based scripts and compare the performance of proposed solutions.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS **1**

ABSTRACT **2**

1. INTRODUCTION **4**

1.1 CHALLENGES OF INDUSTRY 4.0 4
1.2 METAL PARTS MANUFACTURING INDUSTRY AND THE INSTRUMENTS 4
1.3 THE DEMAND FOR PRECISE TOOL LIFETIME DATA 6

2. METHODS FOR PREDICTIVE MAINTNENACE **8**

2.1 PREDICTION METHODS BASED ON PHYSICAL DATA 8
2.2 MACHINE LEARNING ALGORITHMS 11
2.3 PREDICTIVE MODELS BASED ON MACHINE LEARNING ALGORITHMS 14
2.4 PREDICTIVE MODELS BASED ON HYBRID MODELS 16

3. IMPLEMENTATION SOLUTION ON PYTHON **17**

3.1 DATASET ANALYSIS 17
3.2 DATASET PREPARATION FOR IMPLEMENTATION 20
3.4 NEURAL NETWORK 28
3.5 HYBRID MODEL 29
3.6 USED INSTRUMENTS 30

4. PERFORMANCE ANALYSIS **32**

4.1 METRICS 32
4.2 RMSE RESULTS 33

5. CONCLUSIONS **36**

5.1 SUMMARY OF THE RESEARCH 36
5.2 FURTHER DEVELOPMENTS 37

1. INTRODUCTION

1.1 Challenges of Industry 4.0

The term Industry 4.0 was introduced in recent years due to the revolutionary changes in paradigms of production and manufacturing. Within this phenomenon, various technologies are combined, including Internet of Things (IoT), artificial intelligence (AI) and Cloud technologies, which are enabling factors for the development of various industries including mechanical parts production. The new technologies enable to develop and increase productivity of manufacturing sector. However, in order to adapt to dramatic changes in the sector the current processes of production need to be reconsidered and optimized.

The emerging paradigm will replace the traditional one in which the control of processes is entrusted to manually written codes for PLC (programmable logic controller) then interpreted by the operators with standardized procedures. New developments in certain fields such as applied mathematics and software statistical learning, with the increasing availability of open source tools, offer an incentive to increase data collection in production processes and their use in models based on data collected with new technologies.

The global computer numerical control machines market size is projected to reach USD 132.93 billion by 2030, registering a CAGR of 10.2% from 2022 to 2030, according to a new study by Grand View Research, Inc. The market demand is expected to reach over 2,800 thousand units by 2030. The growth can be ascribed to an increase in the need for automated and high-precision computer numerical control (CNC) machines, which is anticipated to drive market growth over the forecast period.[1] New machineries enable to collect and use data to increase efficiency and create opportunities in the sector. In order to keep in pace with technical capabilities the need for development

1.2 Metal parts manufacturing industry and the instruments

Metal parts manufacturing is a broad term that can refer to the process for metal parts manufacturing such as bending, cutting, assembling, shaping, or molding metals into the desired structure. This process a value-addition process involving the creation of parts, machines, and other structures from raw materials.

In metal parts manufacturing, instead of assembling ready-made components to get the end product, you produce the end product from the raw or semi-raw material. The process employs sheets, rods, billets, and bars of stock metal to produce the new metal

[1] Computer Numerical Control Machines Market Size, Share & Trends Analysis Report By Type (Laser Machines, Milling Machines, Laser Machines), By End Use, By Region, And Segment Forecasts, 2022 - 2030

parts. The total annual revenue from the fabricated metal products manufacturing sector amounted to 86.6 billion euros, with a relevant contribution by the manufacturing of metal structures and parts of structures sector (13.5 billion euros).[2]

Metal parts manufacturing varies significantly for different types of metals. Each manufacturing type has its strength and compatibilities. The most popular manufacturing type is milling.

In metal works, Milling is a machining process (Figure 1) which is performed with a rotary cutter with several cutting edges arranged on the periphery of the cutter. It is a multiple point cutting tool which is used in conjunction with a milling machine. This process is used to generate flat surfaces or curved profiles and many other intricate shapes with great accuracy and having a very good surface finish. Milling machines are one of the essential machines in any modern machine shop for several industries, whether it is aerospace, mechanical engineering, mechatronics engineering and etc.

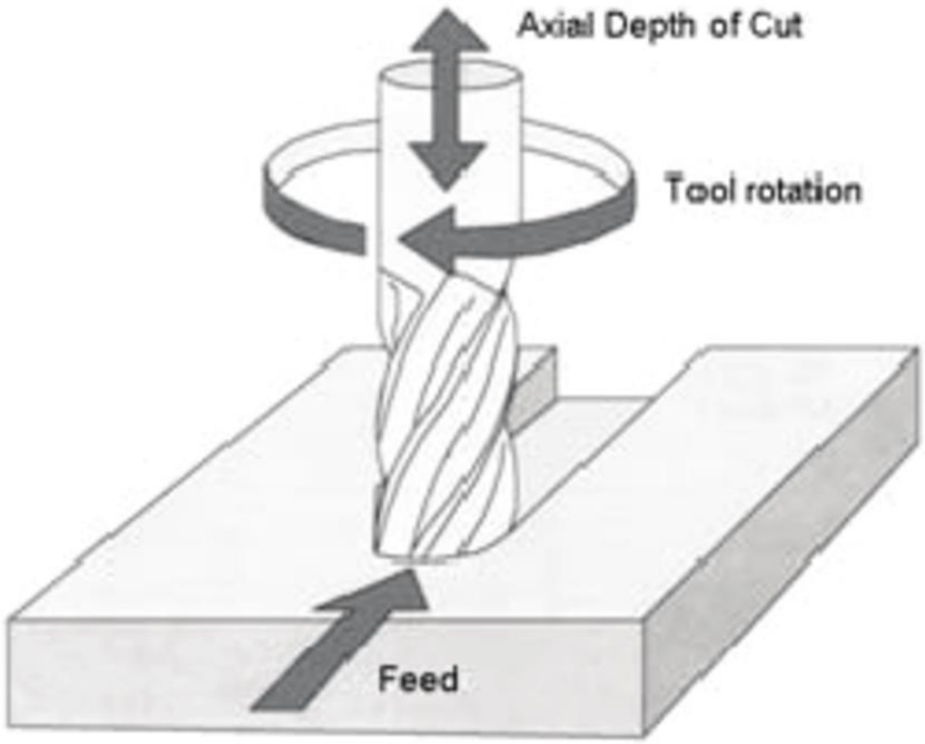


Figure 1 The milling operation using milling tool

In the context of machining, a cutting tool or cutter is typically a hardened metal tool that is used to cut, shape, and remove material from a workpiece by means of machining tools as well as abrasive tools by way of shear deformation. Most of these tools are designed exclusively for metals. There are several different types of single edge

[2] Statista website

cutting tools that are made from a variety of hardened metal alloys that are ground to a specific shape in order to perform a specific part of the turning process resulting in a finished machined part. Single edge cutting tools are used mainly in the turning operations performed by a lathe in which they vary in size as well as alloy composition depending on the size and the type of material being turned. These cutting tools are held stationary by what is known as a tool post which is what manipulates the tools to cut the material into the desired shape. Single edge cutting tools are also the means of cutting material performed by metal shaping machines and metal planning machines which removes material by means of one cutting edge.

1.3 The demand for precise tool lifetime data

The constant contact of the tool with part under high temperatures and pressure cause tool deterioration and creates need for constant maintenance of the instrument. In order to prolong tool life there are special layer of tool flank upon the core metal which can withstand harsh conditions. However, it also can be deteriorated and after wearing out, tool or part is in danger due to the instrument break if maintenance would not be done on time. Tool life data is crucial for managing maintenance works of the tool. Due to the fact, that early tool maintenance (change or repair) cause higher economic cost because of extra labor, maintenance expenditures and during the time of maintenance machine would be idle and, which is also ineffective management of resources. If we consider another scenario when the operator misses tool maintenance time and continue operation even if the tool wear is already exhausted this may lead to even more worse outcomes as tool complete lost, breaking the manufacturing part and even damages to the machine itself.

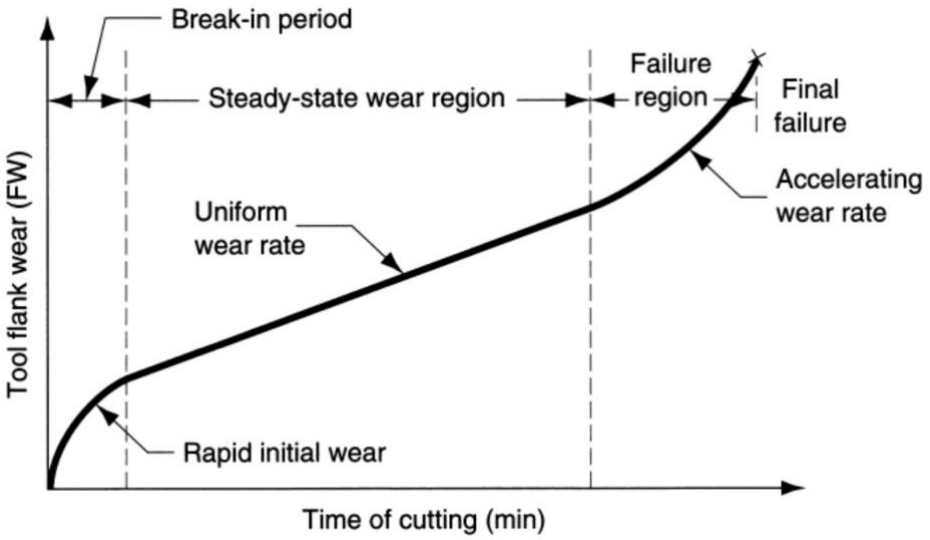


Figure 2 Typical stages of tool wear in machining (Vaughn, 1966)

As indicate in Figure 2 the best moment for tool maintenance is the moment right before the Failure region. In order to meet this goal several research has been done and several approaches emerge and according to literature[3] there are three main types of predictive maintenance

Figure 3 Predictive maintenance classification[4]

[3] F. Trojan, Proposal of Maintenance-types Classication to Clarify Maintenance Concepts in Production and Operations, 2017.

[4] APVV-16-0488 - Innovative system for testing logistic processes by using simulation and emulation

Run-until-Failure method involves intervening only when a failure occurs, therefore it is not suitable in processes where you want to maximize the uptime of the machines and processes require low tolerances

Preventive maintenance method it is scheduled approach based on physical data and sustain maintenance before expected failure, the drawback of this approach is the possible miscalculation if the conditions change and the high cost to sustain.

Predictive maintenance method is based on the Data analysis and behavior prediction using all available data and instruments. It is gaining popularity even if the complexity of this approach is the highest. However, the benefits which offer this way is overcome the cost to implement which goes to sunk costs.

Maintenance methods have evolved over time in parallel with technological progress. In fact, starting from the first preventive maintenance models, an attempt was made to improve performance in terms of reducing operating costs, mainly due to tool costs, and minimizing machine downtime.

2. METHODS FOR PREDICTIVE MAINTNENACE

2.1 Prediction methods based on physical data

Mostly for predict the exact moment for tool change is the physical data models which calculate data using empirical values and constant variables derived by years of practicing and observation. The most popular ones are Archard model and Taylor models which is developed in 20th century. Currently, experienced and professional operators rely on these models to manage their work and tool maintenance schedules.

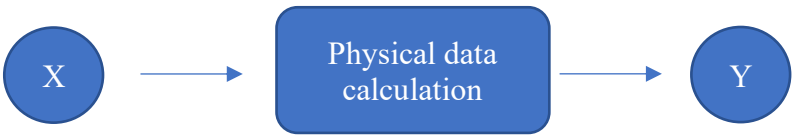


Figure 4 Physical calculation model x-input, y-output

The Archard model

Modelling of wear processes is not limited to metal cutting applications. In the 1950s, British engineer John F Archard developed an empirical model for the rate of

abrasive wear between sliding surfaces based on deformation of the asperity, or roughness, of the surfaces.His equation is:

$$\mathbf{Q= KWL / H.}$$

Equation 1 Archard Model Equation

Here, **Q** is the wear rate, **K** is a constant wear coefficient, **W** is the total normal load, L is the sliding distance of the surfaces, and H is the hardness of the softer of the two surfaces. The model basically states that the volume of material removed due to abrasive wear is proportional to friction forces.

However, the Archard model does not describe tool wear phenomena, but rather predicts the progression rate of wear over time. The model includes the influences of the speed with which the two surfaces interfere with each other, mechanical load, surface strength, material properties and wear coefficient.

Nonetheless, it should be noted that the Archard model was not developed for application at the high speeds common in metalworking, and it does not include the effect of temperature on the wear processes. Both surface strength and wear coefficient will change in response to the 900 deg. Celsius temperatures generated in metal cutting. As result, the Archard model alone does not sufficiently describe tool life in metal cutting.

The Taylor model

In the early 1900s American engineer FW Taylor developed a tool life model that included factors relevant to metal cutting. Taylor observed that increasing depth of cut had minimal effect on tool life. Increasing feed rate had somewhat more effect, while higher cutting speeds influenced tool life the most. This prompted Taylor to develop a model focused on the effect of varying cutting speeds. The equation for Taylor's basic model is

$$\mathbf{vC * Tm = CT}$$

Equation 2 Taylor Model Equation

where **vC** is cutting speed, **T** is tool life, and m and **CT** are constants with **CT** representing the cutting speed that would result in a tool life of one minute.

Taylor also observed that tool wear typically accelerates at the beginning of an operation, settles into a steady but slower rise in a second phase, and finally enters a third and final phase of rapid wear until the end of tool life. He designed his model to represent the length of time between phases two and three.

As a result, Taylor's model does not apply at lower cutting speeds in which workpiece material adheres to and builds up on the cutting edge, affecting the quality of the cut and damaging the tool. Also outside the model's scope are cutting speeds high enough to promote chemical wear. The low- and high-speed wear modes share the characteristics of unpredictability – wear resulting from adhesive or chemical mechanisms can occur either quickly or slowly. The Taylor model is based on the second phase of tool life, namely steady and predictable abrasive wear.

$$T = K * v^{R} * fS * aW * \mathrm{VB_{MAX}}^{Z}$$

Equation 3 Taylor extended model

Where :

• T is the useful life of the tool calculated in t0, expressed in [min];

• K, R, S, W, and Z are empirical constants to be estimated. They will also be called with the term "coefficients", as they represent the coefficients in the model of multiple regression;

• v is the cutting speed expressed in [mm / min];

• f is the feed rate, called feed in the dataset, expressed in [mm / rev];

• a is the depth of cut, denominated DOC in the dataset, expressed in [mm];

• VBmax is the maximum flank wear width that can be measured in relation to activity time T, expressed in [mm].

For the analyzes presented in chapter 3.3, the extended Taylor equation for rotating tools was used, which includes all the machining parameters present in the dataset. The equation has the following form:

$$VB(t) = e^{\frac{\ln(t) - lnK - Rlnv - Sln(f) - Wln(a)}{Z}}$$

Equation 4 Taylor model for milling tool

The original Taylor model concentrates on the effects of cutting speed and is valid if depth of cut and feed do not change. After depth of cut and feed are established, speed is manipulated to modify tool life.

Further experiments led to development of an extended Taylor tool life model equation that included more variables and consequently was more complex. The equation also includes a variable that accounts for the rake angle of the tool, as well as constants for various workpiece materials. Despite the additional factors, this model is most accurate when changing one cutting condition at a time. Altering several conditions simultaneously can produce inconsistent results.

Also, the original Taylor model was unable to fully account for the geometric relationship of the cutting tool to the workpiece. A cutting edge can be engaged in a workpiece in an orthogonal orientation (perpendicular to the direction of feed), or obliquely (at a rake angle relative to the feed direction). And, a cutting edge is considered “free” if its corners are not involved in cutting and “non-free” when the tool’s corner is

engaged in the workpiece. Free orthogonal or free oblique cuts are rarely present in modern metal cutting, so their relevance is limited. Taylor's extended equation added a variable for cutting edge rake angle, but no allowance was made for corner engagement of the tool.

The Taylor model has shortcomings when viewed in hindsight from today's level of metal cutting technology and complexity. However, over its long history the Taylor model has been an excellent basis for tool life predictions and under certain conditions still provides valid tool life data.

Even if these models are widespread among technicians and professionals there a are some drawbacks, such as: constantly monitoring of the tool, difficulty of the process automation. With the increasing rate of production and automation of production processes Industry 4.0 paradigm offers several solutions to modernize tool maintenance procedures and adapt old techniques to the new reality.

2.2 Machine learning algorithms

According to Arthur Samuel Machine learning is defined as the field of study that gives computers the ability to learn without being explicitly programmed. Arthur Samuel was famous for his checkers playing program. Machine learning (ML) is used to teach machines how to handle the data more efficiently. Sometimes after viewing the data, we cannot interpret the extract information from the data. In that case, we apply machine learning. With the abundance of datasets available, the demand for machine learning is in rise. Many industries apply machine learning to extract relevant data. The purpose of machine learning is to learn from the data. Many studies have been done on how to make machines learn by themselves without being explicitly programmed. Many mathematicians and programmers apply several approaches to find the solution of this problem which are having huge data sets.[5]

Machine learning problems divided in two main categories Regression and Classification:

Classification problem is the task of approximating a mapping function (f) from input variables (X) to discrete output variables (y). The output variables are often called labels or categories. The mapping function predicts the class or category for a given observation.

- A classification problem requires that examples be classified into one of two or more classes.
- A classification can have real-valued or discrete input variables.
- A problem with two classes is often called a two-class or binary classification problem.
- A problem with more than two classes is often called a multi-class classification problem.

[5] Mahesh, Batta. (2019). Machine Learning Algorithms -A Review. 10.21275/ART20203995.

- A problem where an example is assigned multiple classes is called a multi-label classification problem.

It is common for classification models to predict a continuous value as the probability of a given example belonging to each output class. The probabilities can be interpreted as the likelihood or confidence of a given example belonging to each class. A predicted probability can be converted into a class value by selecting the class label that has the highest probability.

Regression problem is the task of approximating a mapping function (f) from input variables (X) to a continuous output variable (y). A continuous output variable is a real-value, such as an integer or floating-point value. These are often quantities, such as amounts and sizes.

- A regression problem requires the prediction of a quantity.
- A regression can have real valued or discrete input variables.
- A problem with multiple input variables is often called a multivariate regression problem.
- A regression problem where input variables are ordered by time is called a time series forecasting problem.

Because a regression predictive model predicts a quantity, the skill of the model must be reported as an error in those predictions.

Machine Learning relies on different algorithms to solve data problems. Data scientists like to point out that there‟s no single one-size-fits-all type of algorithm that is best to solve a problem. The kind of algorithm employed depends on the kind of problem you wish to solve, the number of variables, the kind of model that would suit it best and so on. There are several types of machine learning algorithms

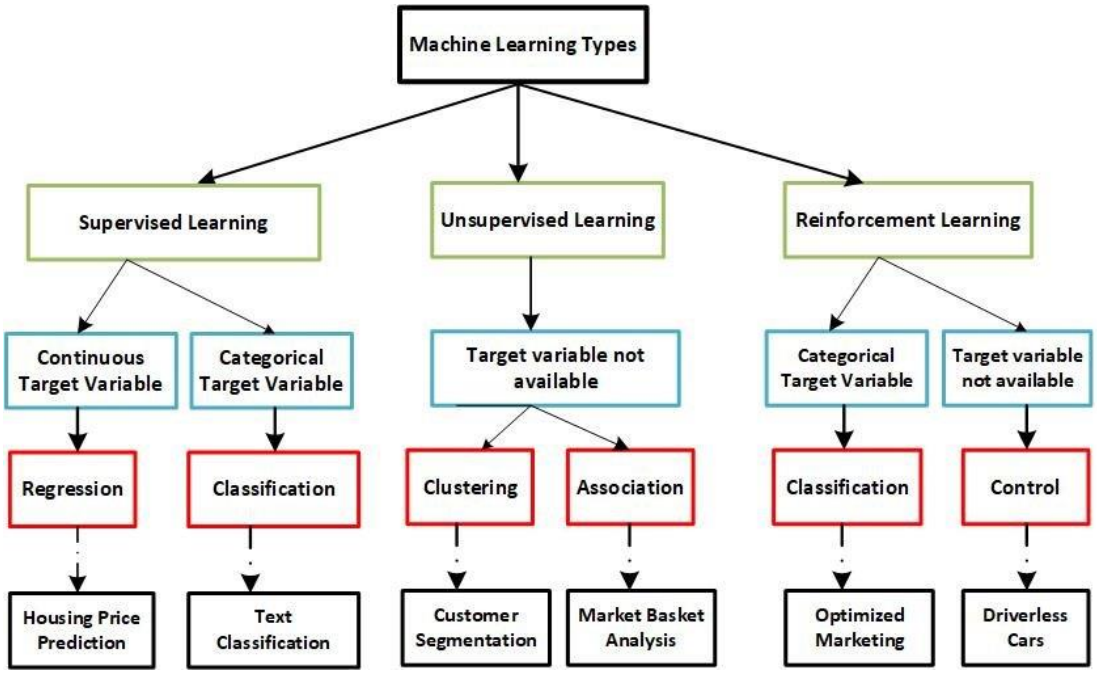


Figure 5 Machine learning algorithms[6]

Supervised Learning - is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labelled training data consisting of a set of training examples. The supervised machine learning algorithms are those algorithms which needs external assistance. The input dataset is divided into train and test dataset. The train dataset has output variable which needs to be predicted or classified.

Unsupervised learning - denotes how a network can studyto signify some input designs in a method that reproducesthe numerical arrangement of the total gathering of inputdesigns or patterns. It is a machine learning charge ofsurmising a function to define the hidden structure from theunlabeled data. This is nothing but the learning algorithms thedo not contain any labels to supervise the learning/training.In the algorithm, a large number of data and characteristic ofeach observation are provided with inputs but not with thedesired output. Unsupervised learning is usually employed(e.g. clustering) to separate the images into two sets orclusters based on some inherent features of the pictureslike color, size, shape etc.

Reinforcement Learning (RL) - This type of machine learning is motivated by a behaviorist psychology. Reinforcement Learning (RL) learnsby interconnecting with its

[6] Unsupervised Learning Based On Artificial Neural Network: A Review October 2018 IEEE International Conference on (CBS)

surroundings or environment. AReinforcement Learning manager or agent learns from the significances of its activities, than from being clearly trained and it chooses its actions to base its past information and alsoby novel choices, which is basically a trial and error learning technique. This is different from classic supervised learning, since precise input/output data sets are not presented. Instead, the emphasis is on the presentation, which comprises findingan equilibrium between exploration (of the uncharted part) plus exploitation (of current data)

2.3 Predictive models based on machine learning algorithms

The availability of information depends on the pre-existing maintenance method on which the ML model is to be developed. In the case of **Run-to-Failure** approach it is possible to adopt a supervised algorithm thanks to the availability of data deriving from a cyclic process. In fact, a supervised algorithm requires a dataset composed of n iterations of a pair $\{x_i, y_i\}$ of observations that contain the information of each i-th iteration of the process. The first term, xi, is a vector that contains the values of the variables under consideration of the process, while y_i is the vector of the outputs of such variables. If y assumes continuous values, there is a regression problem, as in the case of the prediction of the Remaining Useful Life (RUL). While for discrete values of y there is a classification problem, typical of processes that can assume two or more states of activity. The phases that are recurrent in ML methods are: selection of historical data, also called data acquisition in the Predictive Maintenance area, which aims to obtain the data necessary for the analysis; the data pre-processing phase follows, which includes the transformation, cleaning and reduction of the data itself; after obtaining the data in the desired form, the appropriate model must be selected, which therefore requires a validation phase after the training phase; finally, when the model to be adopted has been chosen, the maintenance phase follows during the whole operational period[7]

Figure 6 Machine Learning model x-input, y-output

Nowadays, approaches based on artificial intelligence (AI) are replacing traditional methods[8]. In fact, model-based approaches require mechanical knowledge of the phenomenon, while statistical ones require advanced mathematical knowledge of the process. While these limits are overcome with AI and a study has shown that it manages

[7] G. Soares, An on-line weighted ensemble of regressor models to handle concept drifts, 2015.
[8] T. P., Carvalho, A systematic review of machine learning methods applied to predictive maintenance, 2019.

to achieve better performance than traditional methods. The most commons are Random Forest, Support Vector Machine and k-fold

The Random Forest algorithms is a type of supervised learning algorithms and can be implemented for both regression and classification problems. The underlying principle is the development of a set of randomized decision trees, which constitute a "forest", and the forecasts are given by an average of the outputs provided by each. They perform well when the number of variables is larger than the number of observations available. In fact, compared to deep decision trees approaches, Random Forest methods avoid overfitting because they work with smaller trees formed by random subsets. These methods are among the most used because compared to other ML methods they allow to generalize the model and provide robust estimates. The main disadvantages are related to the complexity of the method and the computational time required to execute it.

Artificial Neural Network (ANN) are the most studied in recent years for the development of predictive maintenance models for milling machines. An ANN consists of a layer of input nodes and one of output nodes, one or more layers of hidden nodes and connecting weights between the various nodes. The network infers the unknown function that relates the data by adjusting the weights on repeated input and output observations. In the literature there are numerous studies that demonstrate the potential of ANNs, among which are: the greater speed of operation compared to traditional multivariate techniques; better accuracy performance than traditional statistical methods, with the possibility of operating without assumptions on the statistical distribution of the function; ability to capture complex phenomena even in the absence of a prior knowledge[9]. While one of the main limitations of ANN techniques is the lack of transparency of how the network evolves and consequently of how it makes decisions. This is due to the complexity of the network structure necessary to model equally complex phenomena, and as reported the transparency of a model is inversely proportional to its complexity. They have been developed several techniques based on ANN, such as deep learning and the Convolutional Neural Network (CNN). The principle of the first concerns the learning of data on different hierarchical levels, which allows to implement complex functions that map the input data directly on the output. While the CCN technique is a particular class of deep learning that integrates data from sensors that measure different parameters. Although these latter techniques sophisticated devices have excellent performance in certain areas, they require advanced knowledge in data selection.

Support Vector Machine (SVM)s are set of related supervised learning methods used for classification and regression[10]. They belong to a family of generalized linear classification. A special property of SVM is, SVM simultaneously minimize the empirical classification error and maximize the geometric margin. So SVM called Maximum Margin Classifiers. SVM is based on the Structural risk Minimization (SRM). SVM map

[9] J. Lee, A systematic approach for developing and deploying advanced prognostics technologies and tools: methodology and applications, 2007.
[10] V. Vapnik. The Nature of Statistical Learning Theory. NY: Springer-Verlag. 1995.

input vector to a higher dimensional space where a maximal separating hyperplane is constructed.

k-means models are very popular in unsupervised techniques for determining a set of partitions. The goal of this algorithm is to find k partitions in which the related data are grouped, and the different ones are separated. It is easy to implement and has good performance even with large amounts of data, minimizing the variance within the classes and maximizing the one between them. While the main criticalities of the algorithm in question are the difficulty of determining the number of partitions (k) and the sensitivity in the order of data entry [11]

2.4 Predictive models based on hybrid models

There are several models based on the combination of existing solutions, as described above each technique has its own points of strength and weakness in order to achieve the best outcome and use with maximum efficiency all possible solutions there exist Hybrid Machine Learning (HML) models. Hybrid Machine Learning models usually achieves higher efficiency using weight allocation approach.

HML is a progress of the ML work process that perfectly unites different computations, processes, or procedures from equivalent or different spaces of data or areas of usage fully intended to enhance each other. As no single cap fits all heads, no single ML procedure is appropriate for all issues. A couple of strategies that are extraordinary in managing boisterous data anyway may not be prepared for dealing with high-layered input space. Some others could scale efficiently on high-layered input space anyway may not be good for managing sparse data. These conditions are a fair motivation to apply HML to enhance the contender procedures and use one to overcome the deficiency of the others.

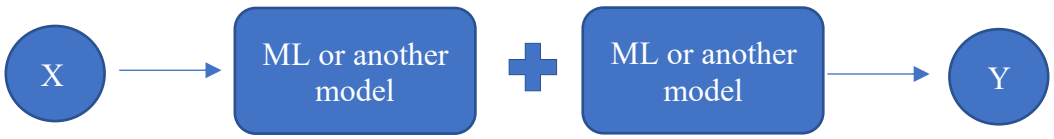


Figure 7 Hybrid Machine Learning model x-input, y-output

[11] G. Hamerly, Partitional clustering algorithms, 2015.

3. IMPLEMENTATION SOLUTION ON PYTHON

The main goal of this implementation is to correctly predict the VB (flank wear) of the tool. In order to achieve this result firstly data need to be prepared and the dataset must be divided to train and test dataframes. After all preparations the predictive models should be trained with train dataset and then this model would be tested on test dataframe.

3.1 Dataset analysis

The analyzes carried out in the thesis research are based on the public Milling dataset (Agogino & Goebel) made available by the Prognostic Center of Excellence (NASA-PCoE)[12]. It contains the values recorded by six sensors throughout the life cycle of 16 tools (cases), under different working conditions identified with 8 scenarios, for a total of 170 processes (runs). Each case is characterized by three processing parameters, which follow the recommendations of the tool manufacturer, and by the type of material processed with fixed dimensions (483 mm x 178 mm x 51 mm). Except for the cutting speed, which remains unchanged at 200 m / min, the other variables are dichotomous and in particular: the feed speed has been set at 0.25 mm / s or 0.5 mm / s; depth of cut (DOC) has been set at 0.75mm or 1.5mm; while the materials of the workpiece are cast iron or 145 stainless steel (stainless steel). Figure below summarizes the set of experiments carried out.

Case	#Runs	Material	Feed	DOC
1	17	Cast iron	0.5	1.5
2	14	Cast iron	0.5	0.75
3	16	Cast iron	0.25	0.75
4	7	Cast iron	0.25	1.5
5	6	Stainless steel	0.5	1.5
6	1	Stainless steel	0.25	1.5
7	8	Stainless steel	0.25	0.75
8	6	Stainless steel	0.5	0.75
9	9	Cast iron	0.5	1.5
10	10	Cast iron	0.25	1.5
11	23	Cast iron	0.25	0.75
12	15	Cast iron	0.5	0.75
13	15	Stainless steel	0.25	0.75
14	10	Stainless steel	0.5	0.75
15	7	Stainless steel	0.25	1.5
16	6	Stainless steel	0.5	1.5

Figure 8 Table of carried experimens

A vertical Computer numerical control (CNC) machine Matsuura MC-510V was used to carry out the machining, with a face milling cutter with six 70 mm inserts. KC710

[12] A. Agogino and K. Goebel, Milling dataset (NASA-PCoE), 2007

inserts were used, characterized by different sequential coatings of titanium carbide, titanium carbonitride and titanium nitride, which give good resistance to cracking and edge wear so that the main wear phenomenon is flank wear.

While the sensors collected data continuously, both during activity and during machine downtime, flank wear (VB) measurements were carried out periodically. In fact, to obtain this measurement it was necessary to remove the insert from the cutter and measure with a special microscope the distance from the cutting edge to the end of the abrasive wear on the side face of the tool. Given the diversity of the machining parameters, cycles with different activity times and number of machining operations have been obtained for each tool. Furthermore, given the discontinuity of the measurements, a fixed maximum flank wear threshold was not respected, obtaining rather heterogeneous maximum values.

The sensors collected data defined on a time series of 9000 units for each process. The measurements were made on the alternating current (smcAC) and direct current (smsDC) of the spindle motor and on the vibrations (vib_table and vib_spindle) and acoustic emissions (AE_table and AE_spindle) on the worktable and on the spindle respectively. All the parameters described above, present in the dataset, are summarized like this:

- *case* - Case number (1-16)
- *run* - Counter for experimental runs in each case
- *VB* - Flank wear, measured after runs; Measurements for VB were not taken after each run
- *time* - Duration of experiment (restarts for each case)
- *DOC* - Depth of cut (does not vary for each case)
- *feed* - Feed (does not vary for each case)
- *material* - Material (does not vary for each case)
- *smcAC* - AC spindle motor current
- *smcDC* - DC spindle motor current
- *vib_table* - Table vibration
- *vib_spindle* - Spindle vibration
- *AE_table* - Acoustic emission at table
- *AE_spindle* - Acoustic emission at spindle

The data collected by the sensors was sent to the processing system through a high-speed data acquisition card with a maximum sampling frequency of 100 KHz. To sample the output data, LabVIEW12 software was used to process the signal generated by the sensors. In addition, the sensor signals, except those of the spindle motor current (both alternating and direct), have undergone a preprocessing process as shown schematically in Figure 9.

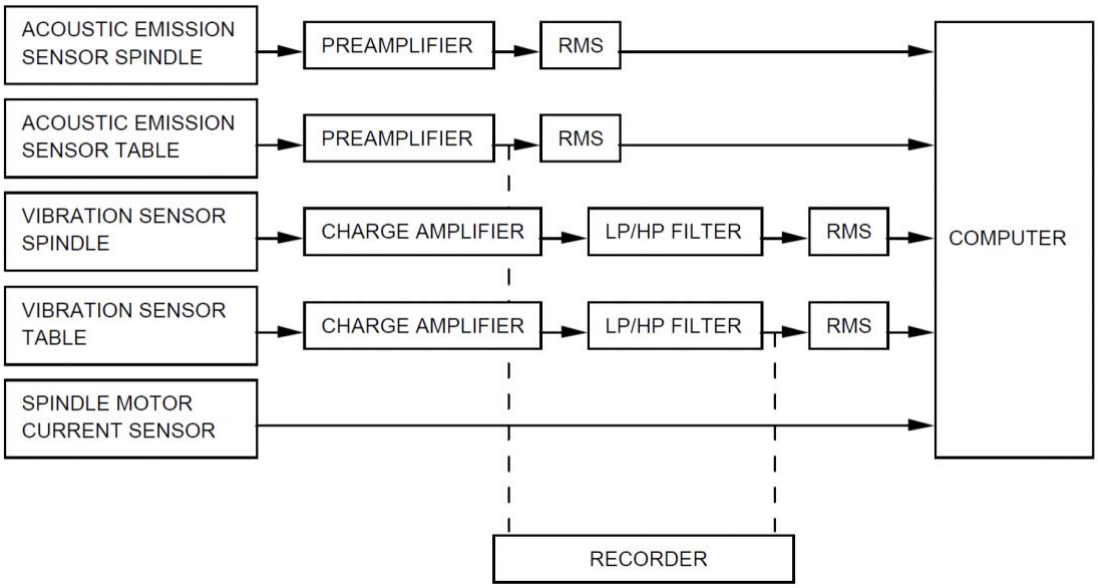


Figure 9 Data collection Setup

For both noise and vibration sensors, the signal has been amplified to meet the minimum threshold requirements required by signal processing equipment. These amplified values were then encoded through the Root mean square (RMS) transformation as reported in equation:

$$RMS = \sqrt{\frac{1}{\Delta T}\int_0^{\Delta T} f^2(t)dt}$$

This step allows to attenuate the signal from the measured oscillations, to make it more accessible to process processing. The RMS is proportional to the energy content of the signal, according to the equation above.

Where ΔT is a constant time interval (fixed at 8.00 ms) and f (t) is the function of the signal coming from the sensors at a given frequency (250 Hz in the case in question). In addition, the data of the acoustic emission and vibration sensors of the worktop were recorded even without the RMS processing. This choice was made to allow comparisons

to be made directly on the raw data. In fact, it can be advantageous to carry out the feature extraction phase on the values supplied directly by the sensors.

3.2 Dataset preparation for implementation

To validate a model and prepare for training and testing it is necessary to divide the dataset into two subsets: the first to train the model (training set) and the second to test it (validation test or test set). This subdivision is necessary to verify the performance of the model without influencing it with the data that have already been observed. By comparing the predictions of the trained model with the training set and with the real values you can evaluate the accuracy of the forecast. With the validated model you can then carry out the future reviews, where the effectiveness can only be measured retrospectively. The validation phase is necessary because with the residual analysis, which is done on the same data used for training the model, there is a risk of evaluating a model affected by overfitting in which the forecasts on unobserved data may present an accuracy. much lower than that observed with residue analysis.

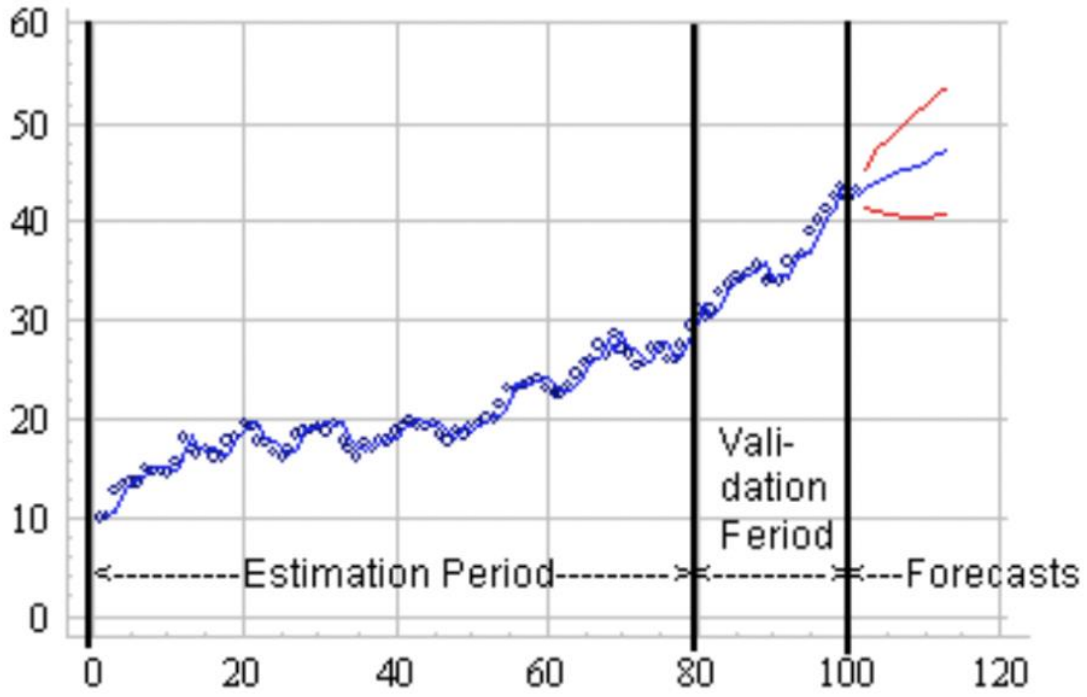


Figure 10 Estimation, Validation, Forecast

The basic method of data validation involves breaking down the dataset into two partitions and performing, as described in the previous paragraph, with one training and the other with validation. However, in the case of samples with little data available, an arbitrary choice of partitions can strongly influence the evaluations. From this consideration comes the cross-validation approach, which involves partitioning the dataset several times in an iterative way. As shown in figure 11, a method that uses this approach involves dividing the dataset into k partitions, using k-1 of them for training and the rest for validation. This process is replicated k times, to have k performance measures with the same dataset. At the end of the iterations, the measures obtained with a statistical operator can be aggregated, the most used being the average, to obtain a more robust measure of the performance of the forecasting model. In fact, the latter no longer depends on the choice of data used for training the method thanks to the iterations that allow you to alternate the data to be allocated to the two partitions several times.

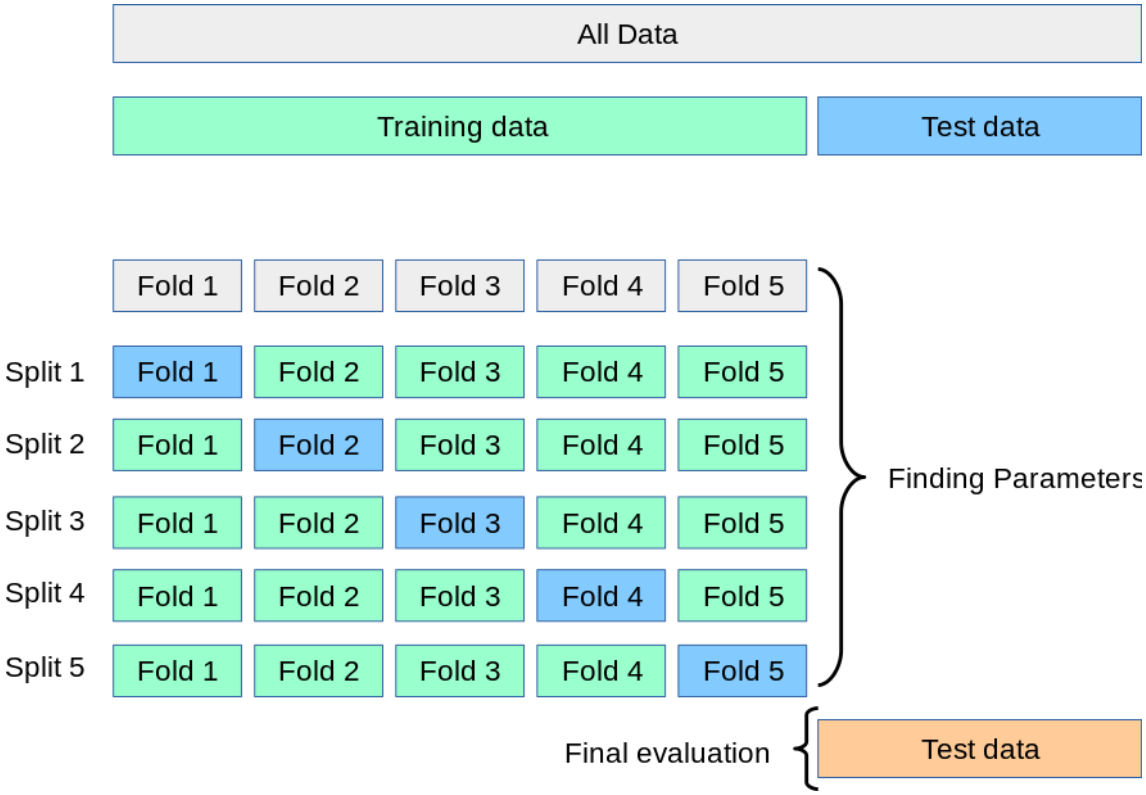


Figure 11 Cross validation[13]

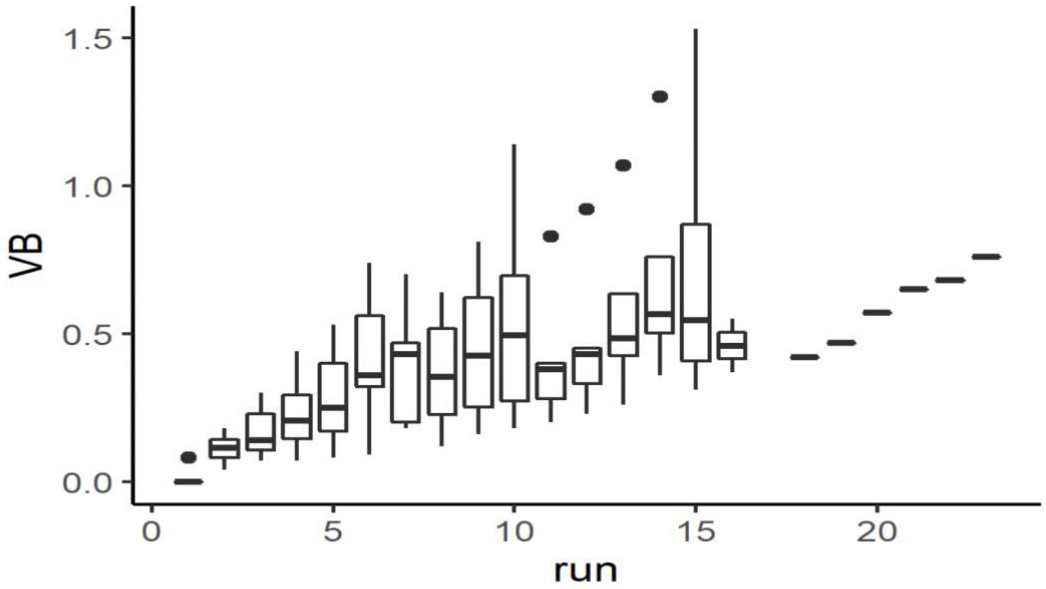
The value of k depends, with an inverse relationship, on the size of the sample. In fact, for small samples k should be large to increase the number of iterations and reduce the bias on the calculated error. Conventionally, k = 5 or k = 10 is used, based on the empirical results of some studies in the literature

[13] Scikit library cross validation representation

Unknown values, several broken data and wrong entries may affect negatively to the training and testing models. In order to avoid these issues before going to implementation the dataset need to be prepared and cleaned from incorrect values this process informally called as “data cleaning”.

For example, there are still some anomalies for the VB values. In particular, the following cases have been removed:

- Case 1: wear level measurements do not respect the condition of increasing monotony. It belongs to the scenario {DOC = 1.5, feed = 0.5, material = cast iron}, whose replication is case 9. The two wear contours should show a similar trend, while it is evident that the measurements of the last points of case 1 differ a lot from the revised trend.
- Case 6: VB values are missing, not allowing to implement any prediction model.



Run (j)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	18	19	20	21	22	23
# tools	9	12	14	10	13	13	9	8	8	6	5	5	4	4	4	2	1	1	1	1	1	1

Figure 12 VB values as the run with the measurements of the dataset

Before analyzing the implemented models, it is interesting to observe the trend of the control variable of the wear process as the runs progress. The box-plots shown in figure 12 have been calculated on the basis of the VB measurements of the different tools for each certain number of runs performed. The goal of this analysis is to demonstrate that the variability of the wear level increases as the runs advance. In fact, in the first processes it follows a constant and almost similar trend for all tools, while when wear reaches an advanced state, its propagation takes on a more random character. From the

aforementioned figure, this phenomenon can be deduced from the increase in the width of the box-plots as the runs increase. However, this trend can be observed only in part due to the limited observations present in the dataset under examination. In fact, the number of tools observed for each level of run performed is variable and makes the comparison between the various runs somewhat approximate. This limitation is more evident for the run numbers that have a single observation, whose box-plots are represented with lines, in which it is not even possible to approximate the variability present. While it is necessary to pay more attention to the runs in the intermediate part (from 11 to 13), where the variability decreases due to the reduced number of observations (about 5 against an average of 10 of the previous runs) and therefore the contraction of their amplitude it is not significant. It can be concluded that the initial hypothesis, concerning the increase in the variability of VB as the runs progress, has been partially confirmed and will be supportive for subsequent analyzes.

Cross validation

As presented before, the cross-validation method approach allows for more robust performance measures by generating different partitions of training set and validation set. In the case in question, this method was applied to the milling dataset. This dataset needs some clarifications. The first concerns the presence of 8 scenarios 13 replicated twice (2 cases 14 were excluded, obtaining a total of 14 cases), therefore it is necessary that in the training set there is at least one case of each scenario to train the models to predict all. possible future scenarios. The second concerns the diversity of the wear phenomenon for the two processing materials, which determine a clearly different tool wear rate. This makes it necessary to enter an equal number of cases for each material to train the models with the same effectiveness for the prediction of the two different wear patterns.

Figure 13 graphically represents the data partitioning carried out with the dataset mill. All the case (tools) present in the dataset constitute the supervised historical data (I_{SUP}) set which is partitioned into the training set (I_{TR}) and test set (I_{TEST}) sets through random extractions. In the figure five tools are extracted for each material, in the following analyzes the models were also tested with four and six extractions per material to evaluate the sensitivity of the models to the size of the training data.

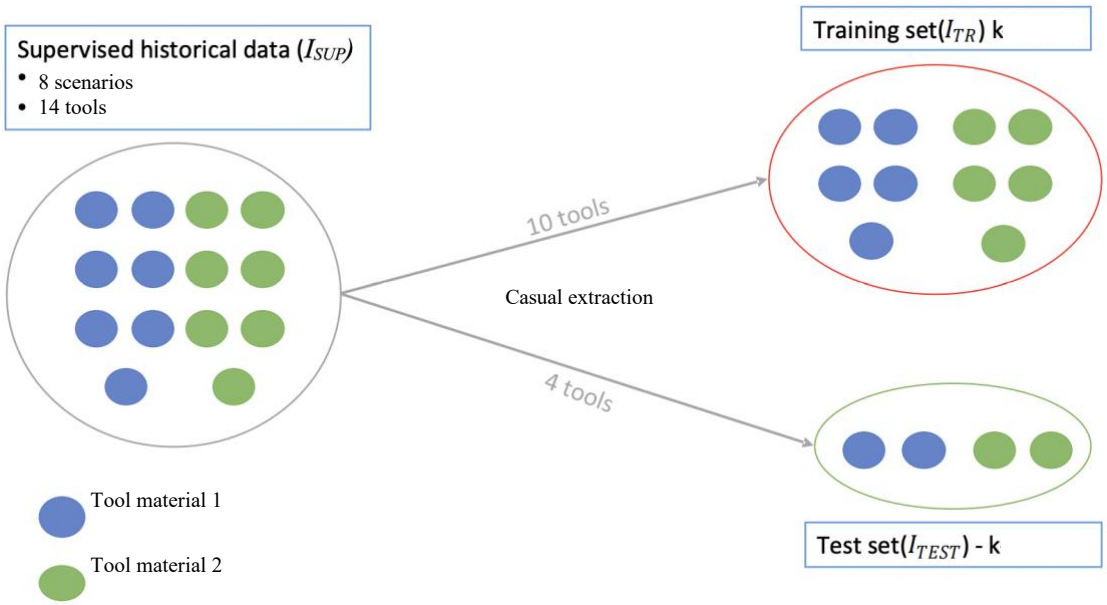


Figure 13 Working principle of Cross validation

Script 1 contains the algorithm implemented to select the cases to be allocated in the two partitions: training and validation. The number of iterations has been fixed at 10 fThe algorithm allows you to choose the number of cases for each material to be used for training, with the variable called "n_case_train_for_m" which can take the values {4; 5; 6}, in order to test the models with different conditions. The script shows only the section for the selection of the houses of the first material, completely similar for the second, which follows three steps:

1. Initialization of the vector that will contain the tools of the training partition with a tool of the material in question extracted randomly.

2. Random extraction of other three tools with the condition that they belong to scenarios that are not already present in the training partition, in order to select a case for each of the four scenarios.

3.If the number of tools required for training is greater than 4, other tools are randomly extracted until the set number is reached. At the end of the selection of tools to be used for training for each material, they are aggregated into a single vector. While the tools to be used for validation are the complementariness of the latter.

```
mill = mill[(mill.case != 1)]

case = mill[['case', 'DOC', 'feed', 'material1']].drop_duplicates()
case_m1 = case[case['material1'] == 1]
case_m2 = case[case['material1'] == 0]
n_case_train_for_m = 6

for k in range(0, 10):
    train_m1 = mill[mill['case'] == 0]
    train_m2 = mill[mill['case'] == 0]

    train_m1 = pd.concat([train_m1, mill[mill["case"] ==
case_m1["case"].sample(1).iloc[0]]])
    train_m2 = pd.concat([train_m2, mill[mill["case"] ==
case_m2["case"].sample(1).iloc[0]]])

    for j in range(2, 5):
        while 1:
            c = case_m1["case"].sample(1).iloc[0]
            parametri = case_m1[case_m1["case"] == c][["DOC", "feed"]]

            if len(pd.merge(train_m1[["DOC", "feed"]], parametri,
how='inner')) == 0:
                break
        train_m1 = pd.concat([train_m1, mill[mill["case"] == c]])

    if n_case_train_for_m - 4 > 0:
        for j in range(1, n_case_train_for_m - 3):
            while (1):
                c = case_m1["case"].sample(1).iloc[0]
                if not (train_m1["case"].isin([c]).any()):
                    break
            train_m1 = pd.concat([train_m1, mill[mill["case"] == c]])

    for j in range(2, 5):
        while 1:
            c = case_m2["case"].sample(1).iloc[0]
            parametri = case_m2[case_m2["case"] == c][["DOC", "feed"]]
            if len(pd.merge(train_m2[["DOC", "feed"]], parametri,
how='inner')) == 0:
                break
        train_m2 = pd.concat([train_m2, mill[mill["case"] == c]])
    if n_case_train_for_m - 4 > 0:
        for j in range(1, n_case_train_for_m - 3):
            while (1):
                c = case_m2["case"].sample(1).iloc[0]
                if not (train_m2["case"].isin([c]).any()):
                    break
            train_m2 = pd.concat([train_m2, mill[mill["case"] == c]])

    train = pd.concat([train_m1, train_m2])
    cases = train["case"].unique()
    test = mill.query("case not in @cases")
```

```
    coef_m1 = pd.DataFrame(columns=list("kswz"))
    coef_m2 = pd.DataFrame(columns=list("kswz"))
```

Script 1 Cross validation and Data cleaning Python script

3.3 Physics based Taylor model

The Taylor model has been implemented as a reference to physics-based methods, as the degradation process analyzed can be modeled with a physical law in mathematical terms. The extended relationship of the Taylor model was chosen because it is widely known and in the case under examination it includes all the machining parameters present in the dataset mill. In fact, the parameters f and a in equation 3 correspond to the attributes of the dataset feed and DOC respectively. While the VB_{max} parameter corresponds to the synthetic notation VB used in the dataset, which refers to the type of flank wear described in paragraph 2.1.

To make predictions of the VB level over time, the inverse relationship of equation 4 was used. As shown in script 2, the dataset values were first transformed with the logarithmic function and then performed a multiple linear regression with the statsmodels.regression.linear_model.OLS function of python library. The result of the regression is the estimate of the empirical coefficients that appear in equation 4, which can be used to predict the level of VB after a certain time of tool activity. These coefficients are related to a specific tool and machining material, which is why it was necessary to set two different regressions for the two materials present in the dataset. Finally, the predictions of VB were made for each material with the relative estimated coefficients.

```
#TAYLOR
    lntrain_m1 = np.log(train_m1[(train_m1.VB != 0) & (train_m1.time
!= 0)][["VB", "time", "DOC", "feed"]])
    lntrain_m2 = np.log(train_m2[(train_m2.VB != 0) & (train_m2.time
!= 0)][["VB", "time", "DOC", "feed"]])
    f_m1 = sm.ols(formula="time ~ feed + DOC + VB",
data=lntrain_m1).fit()
    f_m2 = sm.ols(formula="time ~ feed + DOC + VB",
data=lntrain_m2).fit()


    coef_m1.loc[-1] = np.asarray(f_m1.params)
    coef_m1.iloc[0]["k"] = np.exp(coef_m1.iloc[0]["k"])
    coef_m2.loc[-1] = np.asarray(f_m2.params)
    coef_m2.iloc[0]["k"] = np.exp(coef_m2.iloc[0]["k"])

    est_VB_T = list(range(len(test)))

    for i in range(0, len(test)):
```

```
        if (test.iloc[i].material1 == 1):
            est_VB_T[i] = np.exp((np.log(test.iloc[i].time) -
np.log(coef_m1.iloc[0].k) - coef_m1.iloc[0].s * np.log(
                test.iloc[i].feed) - coef_m1.iloc[0].w *
np.log(test.iloc[i].DOC)) / coef_m1.iloc[0].z)
        else:
            est_VB_T[i] = np.exp((np.log(test.iloc[i].time) -
np.log(coef_m2.iloc[0].k) - coef_m2.iloc[0].s * np.log(
                test.iloc[i].feed) - coef_m2.iloc[0].w *
np.log(test.iloc[i].DOC)) / coef_m2.iloc[0].z)
    RRSE_numpy = np.sqrt(
        np.sum(np.square(np.subtract(test.VB, est_VB_T))) /
np.sum(np.square(np.subtract(test.VB, np.mean(test.VB)))))

    d2 = {'iteration': k, 'method': "Taylor", 'rmse':
mean_squared_error(test.VB, est_VB_T), 'rrse': RRSE_numpy}
    df2 = pd.DataFrame(d2, index={len(result)})
    result = pd.concat([result, df2])
    est_VB_T_train = pd.DataFrame(np.nan, index=range(0, len(train)),
columns=['case', 'run', 'VB'])

    for i in range(0, len(train)):
        est_VB_T_train.iloc[i].case = train.iloc[i].case
        est_VB_T_train.iloc[i].run = train.iloc[i].run
        if train.iloc[i].material1 == 1:
            est_VB_T_train.iloc[i].VB =
np.exp((np.log(train.iloc[i].time) - np.log(coef_m1.iloc[0].k) -
coef_m1.iloc[
                0].s * np.log(train.iloc[i].feed) - coef_m1.iloc[0].w
* np.log(train.iloc[i].DOC)) / coef_m1.iloc[0].z)
        else:
            est_VB_T_train.iloc[i].VB =
np.exp((np.log(train.iloc[i].time) - np.log(coef_m2.iloc[0].k) -
coef_m2.iloc[
                0].s * np.log(train.iloc[i].feed) - coef_m2.iloc[0].w
* np.log(train.iloc[i].DOC)) / coef_m2.iloc[0].z)
```

Script 2 Taylor model Python script

Before proceeding with the predictions of the BV levels it is necessary to analyze the result of the regression used to estimate the coefficients. In fact, to be valid the expected physical relationship for the degradation phenomenon, the regression analysis must have a high statistical significance. The regression results for estimating the coefficients of the first material, which are also representative of the second, are shown in Table 9.5.1. It can be noted that the parameter of the cutting speed among the regressors has been omitted, since, although it is the parameter that most influences the useful life of the tool, in the experiments of the dataset it is kept constant at 200 m / min, not allowing to estimate the impact on the propagation of wear. However, all the other processing parameters are highly significant (the symbol *** in the table corresponds to an $\alpha = 0$) and help to explain the phenomenon under examination. The R2 indicator also confirms that the chosen model has a functional form suitable for the phenomenon of degradation

underlying the trend of the VB level, allowing to explain more than 93% of the variance of the data.

3.4 Neural network

The implementation of machine learning algorithms has made it possible to include data recorded by sensors in the models as well. These data cannot be provided to the models directly as they have been recorded on multiple instants in time for each run. The models chosen are the most popular in the field of predictive maintenance as it emerged in the study of the state of the art in Chapter 2. To these the linear regression model was added to have a baseline on which to compare performance. The functions used are shown in sequence in script 3. In particular:

- MLPRegressor of scikit library Multi-layer Perceptron regressor.

  Scikit-learn (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. Scikit-learn is a NumFOCUS fiscally sponsored project.

- Limited-memory BFGS (L-BFGS or LM-BFGS) is an optimization algorithm

  The BFGS method, an iterative numerical optimization method, is named after its researchers: Broyden, Fletcher, Goldfarb, Shanno. It belongs to the class of so-called quasi-Newtonian methods. In contrast to the Newtonian methods, the Hessian of the function is not directly calculated in the quasi-Newtonian methods, i.e. there is no need to find second-order partial derivatives. Instead, the Hessian is calculated approximately from the steps taken so far.

  There are several modifications of the method:

  L-BFGS (limited memory usage) - used in case of many unknowns.

  L-BFGS-B is a limited memory modification in a multidimensional cube.

  The method is efficient and stable, so it is often used in optimization functions. For example, in SciPy, a popular library for the python language, the optimize function defaults to BFGS, L-BFGS-B.

```
# neural network (NN)-----------------------------------------
-------------------------------------------------

clf = MLPRegressor(hidden_layer_sizes=(100, 100, 100),
max_iter=80, alpha=0.001, solver='lbfgs', verbose=20,
                   random_state = 10, tol=0.001)

y = np.asarray(train['VB'], dtype="float")

x = train.drop('VB', axis=1)
clf.fit(x, y)
est_VB_NN = clf.predict(test.drop('VB', axis=1))

RRSE_numpy = np.sqrt(
    np.sum(np.square(np.subtract(test.VB, est_VB_NN))) /
np.sum(np.square(np.subtract(test.VB, np.mean(test.VB)))))
d2 = {'iteration': k, 'method': "NN", 'rmse':
mean_squared_error(test.VB, est_VB_NN), 'rrse': RRSE_numpy}
df2 = pd.DataFrame(d2, index={len(result)})
result = pd.concat([result, df2])
res = clf.predict(train.drop('VB', axis=1))
est_VB_NN_train = pd.concat([train['case'], train['run'],
pd.Series(res)], axis=1, keys=['case', 'run', 'VB'])
est_VB_NN_train = est_VB_NN_train.fillna(0)
```

Script 3 Neural Network implementation using MLPRegressor(scikit.learn)

3.5 Hybrid Model

As emerged from the analysis of forecasting errors as the runs progress, the accuracy of physics-based and data-driven methods varies over the course of the wear phenomenon. Where, in particular, in the linear sections the former is more accurate and in the non-linear ones the latter. From this result, a hybrid model was created between the two approaches that for each run generates a forecast from the linear combination of the predictions of the single models. In this way, the potential of each method is exploited for the section of the wear trend where it is most effective. The hybrid model developed has the following steps

1. Training of Taylor and NN models individually with I_{TR} training set as shown in paragraphs 3.3 and 3.4;
2. Generation of a set of predictions of VB, for each run (j), on the training set with both single models: VBtaylor, *j* and VB_{NN}, *j* with j = {1, 2, ..., max_i (run_{max}, *i*)} 16, where the

first subscript refers to the model with which the predictions were made e the second indicates the j-th run of which predictions are made for all the tools that present the observation of this run in I_{TR}
3. Compute weights w_j ε [0,1] 18 for each run j, to generate the linear combinations of predictions as reported in the following equation:

$$VB_{hybrid,j}(w_j) = w_j * VB_{Taylor,j} + (1\text{-}w_j) * VB_{NN,\,j}$$

Equation 5 Calculation of Hybrid model

3.6 Used Instruments

Python

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library. Python ranks as one of the most popular programming languages using for machine learning applications artificial intelligence problems among developers[14]

```
import numpy as np
import pandas as pd
import statsmodels.formula.api as sm
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error
from pathlib import Path
from sklearn.neural_network import MLPRegressor
```

Script 4 Used libraries

NumPy

[14] "Stack Overflow Developer Survey 2022". Stack Overflow. Retrieved 12 August 2022.

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors.

Pandas

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals. Its name is a play on the phrase "Python data analysis" itself

Scikit-learn

(formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose

MLPRegressor

Multi-layer Perceptron (MLP) is a supervised learning algorithm that learns a function by training on a dataset, where is the number of dimensions for input and is the number of dimensions for output. Given a set of features and a target, it can learn a non-linear function approximator for either classification or regression. It is different from logistic regression, in that between the input and the output layer, there can be one or more non-linear layers, called hidden layers.

In MLPRegressor was used implemented solver for LBFGS optimization method

The BFGS method, an iterative numerical optimization method, is named after its researchers: Broyden, Fletcher, Goldfarb, Shanno. It belongs to the class of so-called

quasi-Newtonian methods. In contrast to the Newtonian methods, the Hessian of the function is not directly calculated in the quasi-Newtonian methods, i.e. there is no need to find second-order partial derivatives. Instead, the Hessian is calculated approximately from the steps taken so far.

There are several modifications of the method:
L-BFGS (limited memory usage) - used in case of a large number of unknowns.
L-BFGS-B is a limited memory modification in a multidimensional cube.

The method is efficient and stable, so it is often used in optimization functions. For example, in SciPy, a popular library for the python language, the optimize function defaults to BFGS, L-BFGS-B.

4. PERFORMANCE ANALYSIS

In order compare and give KPIs to our work there should be some units to measure each algorithm. Here given possible KPI candidates which can be considered as units of performance.

4.1 Metrics

The first metric that can be adopted is the forecast error (equation 5), which represents the difference between the forecast made and the actual value for each point. A limitation of this metric is the dependence on the scale of values, not allowing to compare forecasts on values of different magnitudes.

$$\text{Error}(e_i) = Y_{i\ \text{effective}} - Y_{i\ \text{calculated}}$$

Equation 5 Error of Forecast

The following are aggregated metrics, which allow you to evaluate the performance of the model over the entire horizon of the validation sample. The first metric of this family is the average error equation 6 which will rarely be used due to its limitations. In fact, by averaging the errors with the respective signs, there is a compensation between them, leading to underestimate the error of the model. Furthermore, this measure does not consider the order of magnitude of the phenomenon being monitored, making it difficult to compare the model in different scenarios.

$$Mean\ Absolute\ Percentage\ Error\ (MAPE) = \frac{1}{N}\frac{\sum_{i=1}^{N}|Y_{i\,effective} - Y_{i\,calculated}|}{\sum_{i=1}^{N} Y_{i\,effective}}$$

Equation 6 MAPE Equation

The following metrics are the most widely used to evaluate the performance of forecasting models and will therefore always be calculated in the analyzes carried out in this thesis research. As has been described, MAPE already has good comparative properties of different models. However, with the same MAPE, models can be presented that make estimates with good accuracy in all points and models that have excellent accuracy in some points and poor in others. To overcome this criticality, quadratic errors are used, which have the property of strongly penalizing the larger errors and penalizing the smaller ones less. The widely known RMSE is based on this principle and presents the following equation:

$$Root\ Mean\ Squared\ Error\ (RMSE) = \sqrt{\frac{\sum_{i=1}^{N}(Y_{i\,effective} - Y_{i\,calculated})^2}{N}}$$

Equation 7 RMSE Equation

4.2 RMSE results

Taylor model results after 10 iteration using different cases on each iteration shown in table below. What we can see is that rmse is case sensitive since cases chosen randomly and relatively to the input data errors can vary.

iteration	method	rmse
0	Taylor	0.011038
1	Taylor	0.043569
2	Taylor	0.105843
3	Taylor	0.245859
4	Taylor	0.124392
5	Taylor	0.105843
6	Taylor	0.011038
7	Taylor	0.245859
8	Taylor	0.225159
9	Taylor	0.073231

Table 1 Taylor Model RMSE results

Neural Network Module

Here given the data for Neural Network model results with the same cases as in Taylor's relative iteration we can observe that relusts are vary from Taylor which means that approaches and result are not the same

iteration	method	rmse
0	NN	0.080063
1	NN	0.160342
2	NN	0.10524
3	NN	0.131427
4	NN	0.263336
5	NN	0.105241
6	NN	0.080062
7	NN	0.131427
8	NN	0.125788
9	NN	0.086289

Table 2 Neural Network RMSE results

Hybrid Model

In this scenario Hybrid model show the best results with given cases and the overall data looks favorable to Hybrid.

iteration	method	rmse
0	Hybrid	0.055812
1	Hybrid	0.055366
2	Hybrid	0.087044
3	Hybrid	0.11939
4	Hybrid	0.134925
5	Hybrid	0.09686
6	Hybrid	0.053738
7	Hybrid	0.089274
8	Hybrid	0.076227
9	Hybrid	0.046579

Table 3 Hybrid model RMSE resultsOverall comparisons

In Figure 12 displayed the overall model performance on a scatter plot where we can see performance comparison of proposed methods:

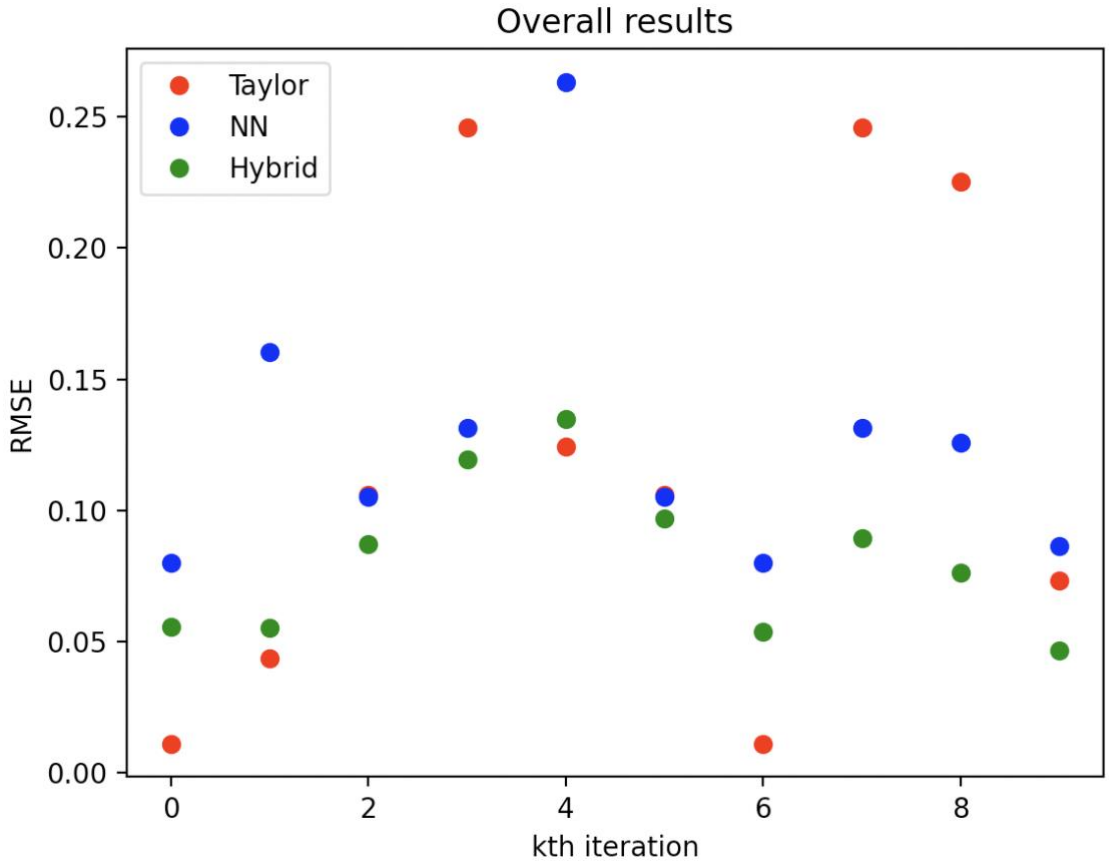


Figure 14 Scatter plot of each iteration of methods RMSEs

iteration	method	rmse
0	Taylor	0.011038
0	NN	0.080063
0	Hybrid	0.055812
1	Taylor	0.043569
1	NN	0.160342
1	Hybrid	0.055366
2	Taylor	0.105843
2	NN	0.10524
2	Hybrid	0.087044
3	Taylor	0.245859
3	NN	0.131427
3	Hybrid	0.11939
4	Taylor	0.124392
4	NN	0.263336
4	Hybrid	0.134925
5	Taylor	0.105843
5	NN	0.105241
5	Hybrid	0.09686
6	Taylor	0.011038
6	NN	0.080062
6	Hybrid	0.053738
7	Taylor	0.245859
7	NN	0.131427
7	Hybrid	0.089274
8	Taylor	0.225159
8	NN	0.125788
8	Hybrid	0.076227
9	Taylor	0.073231
9	NN	0.086289
9	Hybrid	0.046579

Table 4 Overall RMSE comparison

5. CONCLUSIONS

5.1 Summary of the research

The results obtained from the implementation of the models in the previous chapters, respect the theoretical characteristics described for each category of methods for monitoring the wear of the tools analyzed. Each category performed better than the others in certain circumstances, therefore it can be said that it is not possible to identify an optimal method for describing all wear trends, just as there is no single reference model for the scope of forecasts in general

The analyzed category of physics-based methods was found to be the most accurate and robust in the case under examination. In fact, with the Taylor model the best overall forecasting performance was obtained under different conditions. It has the advantage of being able to be implemented even in the absence of sensors on the machine and with few measurements to train the model. In fact, it has been shown that even with

a very limited number of data it is possible to implement a physics-based model that provides a first approximation of the predictions to be made. The main limitation of this method is that it can only be applied to wear phenomena of which a mathematical law is known that describes its trend. Moreover, even if it has obtained the best overall performances, in certain phases of the wear phenomenon a data-driven method can be more accurate.

Finally, the hybrid approach defined by the linear combination of a physics-based and a data-driven method has made it possible to obtain better performance than both single methods under certain conditions. In fact, if a limited number of data is used for training the data-driven method, the physics-based method is preferable individually. While, when the single models obtain similar performances, the hybrid approach allows to significantly increase the overall accuracy and above all to obtain much more robust results. This is possible thanks to the linear combination weights calculated for each machining of the tool that allow you to select the best method for each phase, linear or non-linear, of the wear trend.

5.2 Further developments

The most interesting results obtained from this thesis research are the performances obtained with physics-based methods and with machine learning algorithms and with the related hybrid model. Search to validate the results is to apply the same methods on a dataset with the same monitoring factors of larger size, given the need for the large amount of data needed to train the models. With further research, these results could be validated on another case study, possibly on a different wear phenomenon.

Therefore, it can be done new to use this technique to develop new solutions combining different ANN algorithms and Physics based data to obtain more robust data to use in developing Industry 4.0 paradigm.