UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Head Office: Università degli Studi di Padova

Department of Technical and Management of Industrial Systems

Ph.D. COURSE IN: Mechatronics and Product Innovation Engineering

CURRICULUM: Logistics and Industrial Plants

CYCLE XXXV

Machine learning for Predictive Maintenance

Study of a predictive model based on product quality

Coordinator: Prof.ssa Daria Battini

Supervisor: Prof. Matteo Mario Savino

Co-supervisor: Dott. Ilenia Zennaro

Ph.D. student: Carlo Riccio

ABSTRACT

Predictive Maintenance (PdM) strategy is nowadays the most suitable in terms of production efficiency and cost reduction, aiming to perform maintenance actions when needed, avoiding unwanted failures and unnecessary preventive actions. The increasingly use of 4.0 technologies in industries has allowed the adoption of recent advances in machine learning (ML) to develop an effective Predictive Maintenance Strategy (PMS). Moreover, production efficiency considers not only production volumes in terms of produced pieces or working hours, but also product quality (PQ), that is always a more important parameter, also to detect possible failures. In fact, PQ could be used as a parameter for predict possible failures and deeply affects production costs. In this context, this study aims to develop a Product Performance Based Maintenance framework through ML to determine the optimal PdM strategy based on the desired level of product quality and production performance. The framework is divided into three parts, starting from data collection, through the choice of the ML algorithm, and result analysis. The framework has been applied within the production line of electromechanical components. Results show that the link between the variables that describe the state of operation of the machine and the qualitative parameters of the production process allows to control maintenance actions based on scraps optimization, achieving an improvement in the operation of the machine. Moreover, the application of the model to the production process saves about 50% of the costs for machine downtime and 64% of the costs for scraps.

SOMMARIO

Le strategie di Manutenzione Predittiva (PdM) sono oggi ampiamente utilizzate perché consentono miglioramenti sull'efficienza produttiva e la riduzione dei costi, in quanto prevedono l'esecuzione di azioni di manutenzione solo quando necessario, evitando guasti indesiderati e azioni preventive non necessarie. L'uso crescente di tecnologie 4.0 nell'industria ha consentito l'adozione dei recenti progressi Machine Learning (ML) per sviluppare strategie di manutenzione predittiva (PMS) estremamente efficaci. Inoltre, l'efficienza produttiva tiene conto non solo dei volumi di produzione in termini di pezzi prodotti o di ore lavorate, ma anche della qualità del prodotto (PQ), che è un parametro sempre più importante, anche per rilevare possibili guasti. Infatti, la PQ potrebbe essere utilizzata come parametro per prevedere possibili guasti, e incide profondamente sui costi di produzione. In questo contesto, questa ricerca ha l'obiettivo di creare un framework di manutenzione basato sulla qualità di prodotto attraverso il Machine Learning per determinare la strategia ottimale di manutenzione predittiva in base al livello desiderato di qualità del prodotto. Il framework è suddiviso in tre step, che partono dalla raccolta dei dati, proseguono con la scelta dell'algoritmo di ML, e terminano con l'analisi dei risultati. Il modello è stato testato all'interno di una linea di produzione di componenti elettromeccanici. I risultati mostrano che il legame tra le variabili che descrivono lo stato di funzionamento della macchina ei parametri qualitativi del processo produttivo, consente di impostare le azioni di manutenzione ottimizzando allo stesso tempo il tasso di scarto, ottenendo un miglioramento delle performance della macchina. Inoltre, l'applicazione del modello al processo produttivo consente di risparmiare circa il 50% dei costi per i fermi macchina e il 64% dei costi per gli scarti.

INDEX

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

- AI: Artificial Intelligence
- ANN: Artificial Neural Network
- CNC: Computer Numeric Control
- FIE: Fuzzy Inference Engine
- FPY: First Pass Yield
- I4T: Industry 4.0. Technologies
- IoT: Internet of Things
- LSTM: Long Short-Term Memory
- ML: Machine Learning
- MTBM: Mean Time Between Maintenance
- NBC: Naïve-Bayes Classifier
- NNC: Nearest Neighbor Classifier
- PdM: Predictive Maintenance
- PMS: Predictive Maintenance Strategies
- PQ: Product Quality
- RNN: Recurring Neural Network
- RQ: Research Question
- VP: Vacuum Pump

1 INTRODUCTION

1.1 OVERVIEW

Maintenance, production, and product quality are strictly interconnected concepts.

Maintenance, whose function is increasingly strategic among the activities that directly contribute to the creation of value, got a key role in Industry 4.0. Technologies (I4T).

Within I4T, the data that modern industrial machinery can generate and communicate, are allowing a constant evolution of the Predictive Maintenance Strategies (PMS) to anticipate possible machine failures or shorten the MTBM (Papadopoulos et al., 2021).

The prediction of such behaviors can be achieved with models based on Artificial Intelligence (AI) techniques that, through learning functions may modify maintenance strategies as regards the effective behavior of the production system.

The goal of PMS is not limited just to warrant the continuity of production, minimizing down-time and optimizing stops, but also to keep as much as possible the Product Quality (PQ) (Changchao et al., 2017). According to this concept, certain PQ parameters may be related to machine's operating parameters, whose trends affect the PMS. Thus, also the PQ parameters may be one of the entry data that an AI model can consider in defining PMS.

This work is intended to explore the elaboration of models based on AI techniques that allow to determine the optimal PdM strategy based on the desired level of quality of the product. The goal is the creation of a Product Quality Based Maintenance framework. The implementation of this model involves optimization of the total costs. Therefore, the study evaluates the impact on the total costs, considering both the costs of machine downtime, but also the costs of scraps.

1.2 INDUSTRY AND RESEARCH MOTIVATIONS

Preventive maintenance is a diffused strategy which provides maintenance actions based on a schedule aiming to avoid failures; even if it is an effective strategy, it is not optimal in terms of costs since often it leads to unnecessary replacements or unwanted failures (Florian et al., 2021).

The main goal of predictive maintenance, instead, is to detect coming failures in advance, through the regular monitoring of process, machine, material and product conditions of manufacturing systems, to develop just-in-time maintenance actions; this strategy allows to maintain the availability, quality and safety of the equipment, and to reduce costs related to failures and unnecessary maintenance activities (Krishnamurthy et al. 2005; Mobley, 2011).

The main question that gave rise to the idea behind this study is: "Is it possible, on the basis of the enormous amount of data that modern industrial machines make available, to determine when to carry out predictive maintenance? And if so, is it possible to link predictive maintenance activity to product quality control?"
In order to answer these questions, the present study is focused on the construction of a framework that interprets, through AI techniques, the machine data to establish when it is more appropriate to carry out predictive maintenance, and how this has an impact on product quality.
From the point of view of industrial management, the final objective is the possibility of monitoring and controlling, within the same framework, the continuity of the production flow and the quality rate of the process, simultaneously optimizing the costs of both machine downtime and scraps.

1.3 RESEARCH OBJECTIVES AND QUESTIONS

The research objective of this study is to develop an AI model capable of i) predicting potential machine failures – ii) defining the development of the correct maintenance actions – iii) preventing the potential deviation of quality parameters of the product
The first research question aims to address the feasibility of a model that considers PQ parameters as potential variables for PMS. The basic investigation is focused on the potential relations of machines' data with predictive maintenance indicators and PQ.

> **RQ_1:** How is it possible to consider product quality parameters within the problem of predictive maintenance?

The second research question aims to investigate the potential impact of PMS according to a certain level of PQ. This portion of the research has been conducted by investigating the ability of the model realized to "learn" from the PQ data and to suggest the optimal PMS.

> **RQ_2:** May we improve MTBM (Mean Time Between Maintenance) and decrease maintenance and production costs through the PQ optimization?

The goal of our Research Framework is to incorporate the Product Quality into the management of Maintenance activities and to implement the suitable PMS according to a variable threshold that describe the quality level the production process must guarantee.

1.4 Outline of the Thesis

The present thesis deals with the implementation of a framework that allows to establish when a predictive maintenance action is required according to the machine status and the quality rate of the process.

Chapter 2 (Figure 1) provides the theoretical background on predictive maintenance, artificial intelligence methods and the applications of these techniques to the maintenance area; the aim is to analyze the state of the art in the sectors involved in the research conducted in this thesis, and to identify possible gaps in the literature.

Chapter 3 (Figure 1) shows how the research framework was set up to build a predictive model in which maintenance actions are related to the quality level of the process, and how the developed model is able to answer research questions.

In the Chapter 4 (Figure 1) the case study is described; in particular, it is explained how, starting from a real production process, the model is trained, and how the data generated by the machines are used to suggest the best maintenance action with the aim of optimizing the quality of the final product. The results obtained are analyzed in terms of process performance, maintenance parameters and costs for the company.

Chapter 5 (Figure 1) contains the analysis of the results, highlighting the advantages and limitations of the developed framework, and addressing future developments.

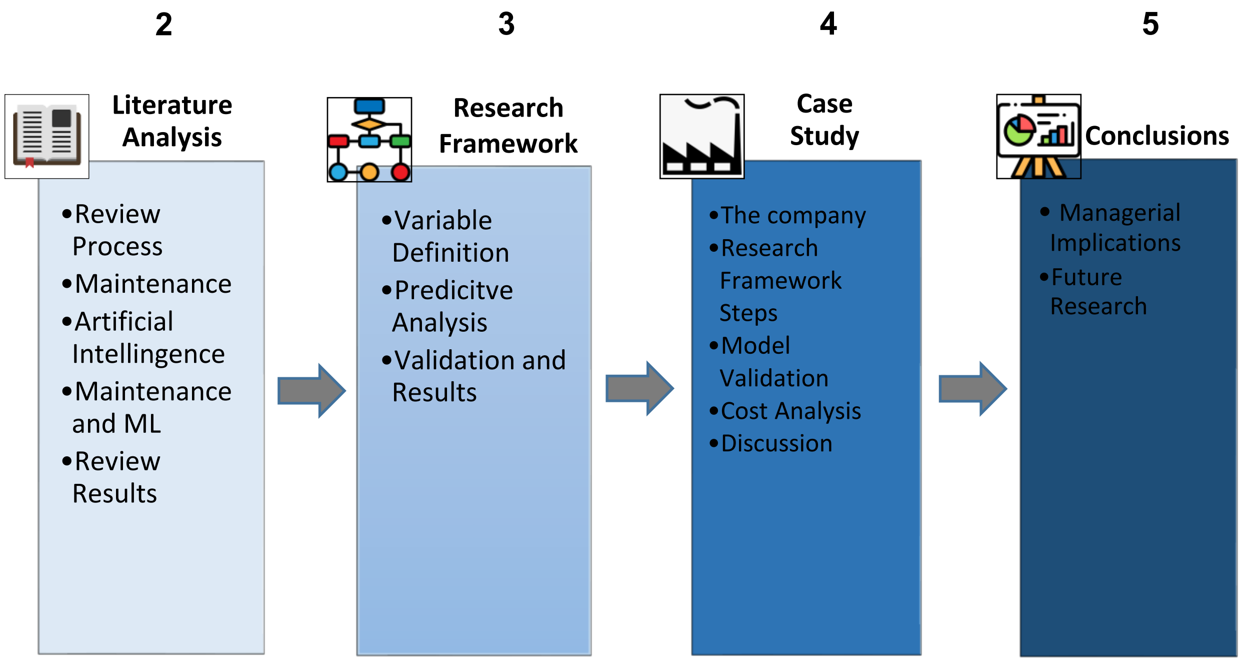


Figure 1: Thesis structure

2 THEORETICAL BACKGROUND

2.1 LITERATURE REVIEW PROCESS

The systematic literature review (Baumeister and Leary, 1997- Webster and Watson, 2002 – Snyder, 2019 – Liberati et al., 2009 – Moher et al., 2009) conducted in this research follows the steps in the guidance by Xiao and Watson (2019), as shown in Figure 2.

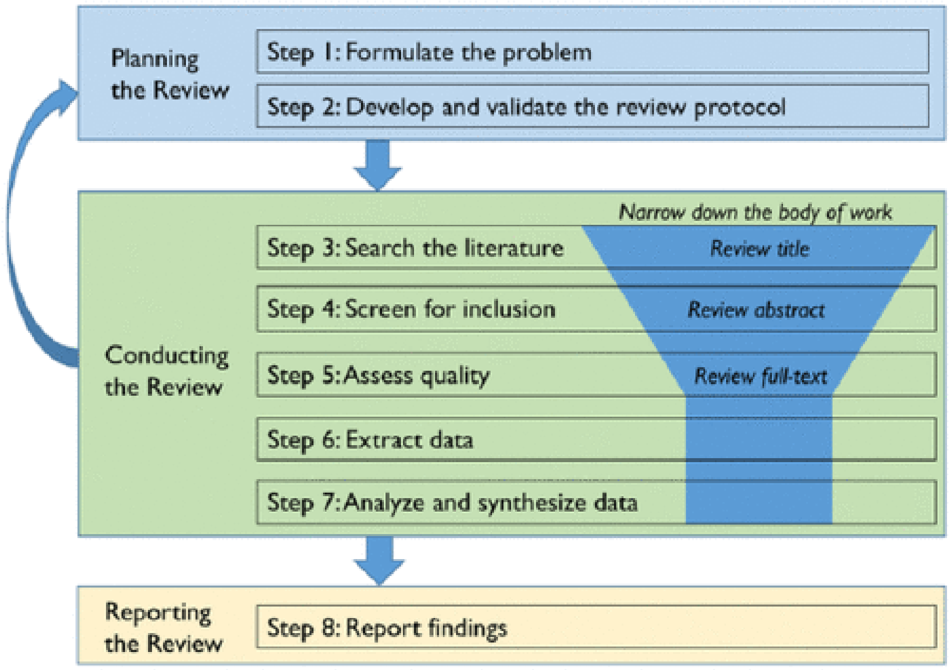


Figure 2: Systematic literature review (Xiao and Watson, 2019)

1) Problem formulation (Kitchenham and Charters, 2007): the research question in Paragraph 1.3 aims to address the feasibility of a model that considers PQ parameters as potential variables for PMS through machine learning techniques, and the possible decreasing of maintenance and production costs through the PQ optimization.
2) Review Protocol (Gates, 2002): the review includes articles, conference paper, books. The protocol includes snowballing, with priority based on number of citations and publication date (from most recent to oldest)
3) Search the Literature: the electronic databases, that constitute the main source of published literature collections (Petticrew and Roberts, 2006), used for searching are Scopus, Google Scholar and ResearchGate; the keywords derives from the research questions and are extended by synonyms, abbreviations, alternative spellings, and related terms (Rowley and

Slack, 2004), and classified according to concept domains (Kitchenham and Charters, 2007) in Table 1:

Table 1: Classification of keywords by topic

Topic			
Maintenance	**Industry 4.0**	**Machine Learning**	**Quality**
Predictive maintenance	Industry 4.0	Artificial Neural Network	Product Quality
Vacuum pump maintenance	Big Data	Nearest Neighbor classifiers	Zero Defect
Thermography	IoT	Naïve Bayes classifier	Product Quality control
Remaining Useful Life	Data collection	Machine Learning	
Condition-based Maintenance	Digitalization	Artificial Intelligence	
Maintenance scheduling		Fuzzy Logic	
CNC Maintenance		Fuzzy model	
Maintenance strategies		Fuzzy sets	
Vacuum pump diagnostic		Deep Learning	
Vacuum pump degradation		Neural Network	
Maintenance Management		Predictive model	
Maintenance activities		Fuzzy rules	
Fault diagnosis		Membership function	
		Backpropagation	

The search was executed with grouped combinations of keywords using the logic AND operator.

4) Screen for inclusion: conducted through the analysis of abstracts with priority based on number of citations and publication date (from most recent to oldest); snowballing is used to identify pertinent work mentioned by the articles and to find all articles that have cited the articles reviewed (Webster and Watson 2002).
5) Assess the quality: conducted through the analysis of full texts with priority based on number of citations and publication date (from most recent to oldest).

6) Extracting the data: conducted as textual narrative synthesis (Lucas et al. 2007). The review has been organized according 3 Paragraphs (Predictive Maintenance Overview, Artificial Intelligence Overview, Maintenance and Artificial Intelligence.
7) Analyzing and Synthesizing Data: the data has been showed as textual description and summary tables (Okoli and Schabram 2010).
8) Report Findings: the summary of the results is presented in the flow chart (Noordzij et al. 2009) in Figure 3, while Table 2 summarizes how the references in the literature review are classified according to some criteria; the analysis of the review is discussed in Paragraph 2.5.

Table 2: Classification criteria for references in LT

Criteria					
Year of publication	< 2000 15	2000-2015 46	> 2015 33		
Type of publication	Journal 35	Conference 49	Book/Technical document 10		
Topic	Maintenance 31	AI and ML 38	I4T 11	Quality 9	Other 5

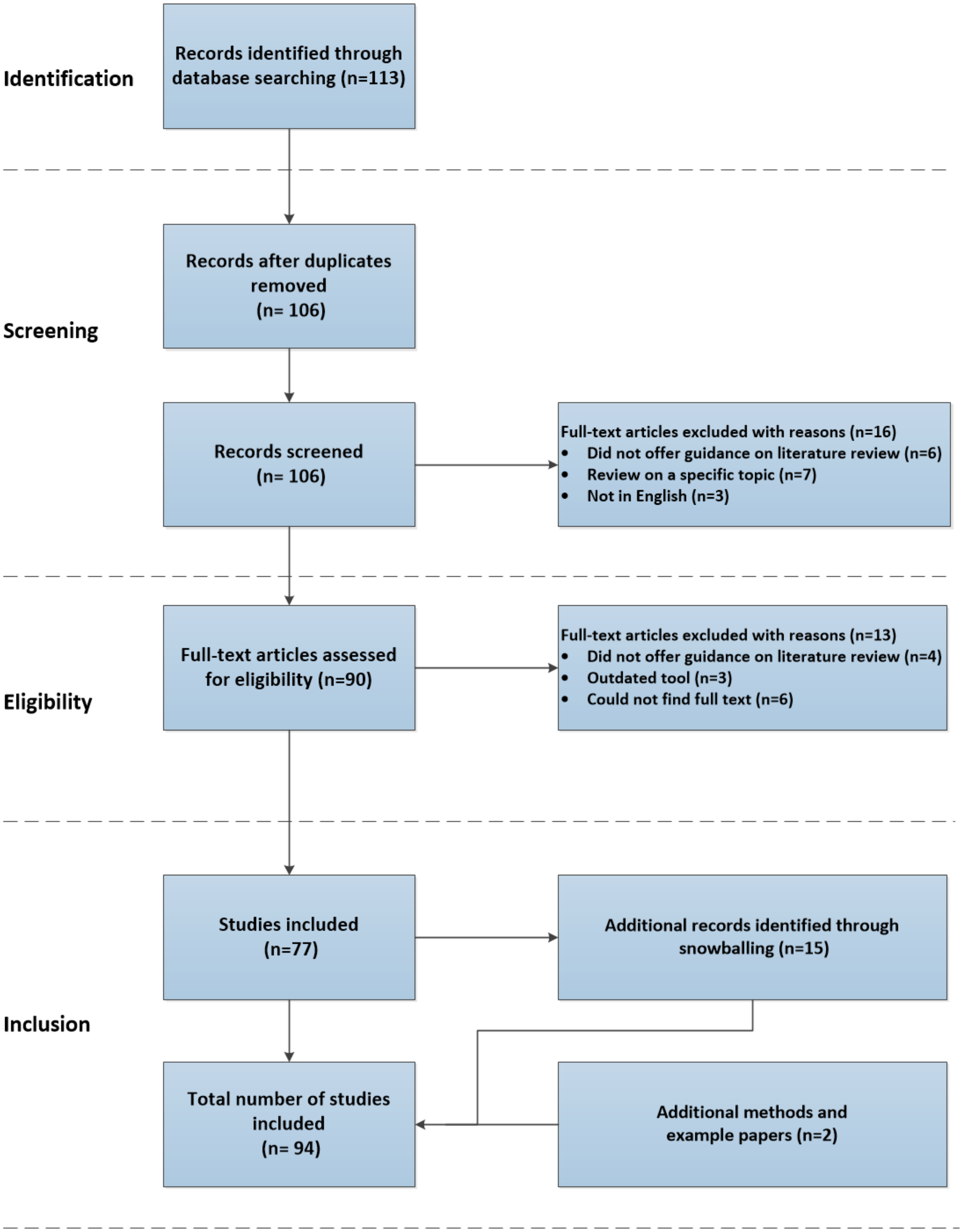


Figure 3: Review results

2.2 PREDICTIVE MAINTENANCE OVERVIEW

Preventive maintenance is a diffused strategy which provides maintenance actions based on a schedule aiming to avoid failures; even if it is an effective strategy, it is not optimal in terms of costs since often it leads to unnecessary replacements or unwanted failures (Florian et al., 2021). The main goal of predictive maintenance, instead, is to detect coming failures in advance, through the regular monitoring of process, machine, material and product conditions of manufacturing systems, to develop just-in-time maintenance actions; this strategy allows to maintain the availability, quality and safety of the equipment, and to reduce costs related to failures and unnecessary maintenance activities (Krishnamurthy et al. 2005; Mobley, 2011).

For doing this, predictive maintenance uses actual operating condition of equipment, material and systems to improve production tasks. Traditional predictive maintenance approaches use a combination of most cost-effective tools, like vibration monitoring, process parameter monitoring, thermography, tribology and visual inspection in order to monitor the actual operating parameters of critical plant systems and to detect potential failures. These manufacturing data are used to plan all maintenance activities on an as-needed base.

According to Carnero (2006), predictive maintenance can be disaggregated into two specific sub-categories:

- Statistical-based Predictive Maintenance, in which the information generated from all stoppages are used to develop statistical models to estimate the Remaining Useful Life (RUL) of the monitored component and thus enables the developing of a preventive maintenance plan.
- Condition-based Predictive Maintenance, which is related to the examination of wear process of mechanical components, and it is supposed that the wear process is related to changes in the machine's behaviour and may cause a mechanical failure.

In both cases, the main requirement for an effective predictive maintenance is enough data from manufacturing process. Data availability is often the main drawback of PMS implementation in manufacturing systems. The greater is the amount of data, the higher will be the accuracy of the health status of the system and the prediction of pending failures. Through the use of integrated sensors, predictive maintenance avoids unnecessary replacement of the equipment, reduces machine downtime, identifies the main cause of the error and, therefore, saves costs by improving efficiency. Unlike conventional preventive maintenance, predictive maintenance planning activities are based on data collected by sensors and analysis algorithms (Wu et. al. 2007; Frontoni et. al., 2017). PMS can be divided into three key phases (Martin, 1994; Jardine, Lin, and Banjevic, 2006): data acquisition, usually carried out automatically, data processing, in which the dataset is cleaned and analysed, and the maintenance decision-making phase, in which maintenance action are planned.

2.3 ARTIFICIAL INTELLIGENCE OVERVIEW

The term "Artificial Intelligence" (AI) describes those systems that show intelligent behavior by analyzing their environment and performing actions, with a certain degree of autonomy, to reach specific goals. AI-based systems may consist of software that operates in the virtual world (e.g., voice assistants, image analysis software, search engines, speech and facial recognition systems); or incorporate AI into hardware devices (for example in advanced robots, self-driving cars, drones or Internet of Things applications)" (Sarzana of S.Ippolito & Nicotra, 2018). It was coined in 1956 by mathematician John McCharty of Stanford University, but it is only today that it became more popular, thanks to the rapid growth of data and the wide availability of computer systems capable of processing the same faster and more accurately than humans can do. The birth of AI stems from the goal of creating a human-like mind exhibited by a computer, robot, or other machine. These tools have the ability to perform people's own cognitive processes, which can be synthesized into:

- comprehension: that is, the ability of the AI to enable a machine to identify figures, patterns, texts, videos, sounds and obtain information by simulating cognitive capacity to correlate data and events;
- reasoning: that is, the ability of systems to link the various information collected through precise mathematical algorithms and in an automated way;
- learning: that is the specific functionalities of the systems that allow the analysis of the data of input and their correct restitution in output;
- interaction: means the way in which AI functions in relation to its interaction with humans. In this field, Natural Language Processing systems have achieved a remarkable advance, technologies that allow man to interact with machines and vice versa by exploiting natural language (Quintarelli, 2020).

At the present time, ANNs help to solve problems in areas such as medicine, biology, sports, production and economics through their specific ability to model, identify and process signals, classify and recognize patterns and predict system's future state (Menanno et al, 2021).

In this research a supervised ANN regulated by the back propagation (BP) of the error, is used for data prediction. In learning stage, a set of training data is used to compute the learning error (Basheer and Hajmeer, 2000). This error is then utilized to correct the network parameters (weights). Thus, if the same data is the input for the network in the next phase, the ANN approximates the goal output.

2.3.1 Machine Learning

In latest years we have seen an increasing spread of the use of machine learning techniques, favoured by the increasing availability of computational resources and data. The applications of machine learning are constantly evolving and begin to support us in many aspects of our daily lives such as autonomous driving systems, spam filters, social media and even in many businesses production processes. Most companies are investing in this sector, which will certainly play a key role in the near future. In this chapter I try to explain what machine learning is and what are the basic knowledge needed to better understand my work.

Machine learning is a subgroup of artificial intelligence that provides machines with the ability to learn and improve themselves automatically through experience, without the need to be explicitly programmed. Using machine learning radically changes the programmer’s approach: if before it was necessary to write very detailed lines of code to show the machine how to behave in every situation, now we have algorithms that can automatically learn and then perform certain actions. In order to acquire this knowledge, algorithms analyse sets of data from which they derive correlations and then real rules. The greater the quantity and quality of the data, the better the predictions made by the algorithm. For this reason, big data plays a fundamental role for the effectiveness of a machine learning algorithm, but they must be managed, prepared and integrated in an appropriate way.

Data used in machine learning can be divided into two categories: labelled data and unlabelled data. Surely having labelled data is better and allows you to achieve better results, but at the same time it also requires a huge manual effort to assign all the labels.

Depending on the problem you have to face and the type of data we have available you can distinguish different types of algorithms, nowadays the most used are three: supervised learning, Unsupervised learning and reinforcement learning.

Supervised learning is the type of algorithms most commonly used and finds application in various fields such as image recognition, text processing, and action recognition. In this case the training is done using labelled data. It is necessary to provide a sufficiently large and varied number of examples to allow the algorithm to generalize the classification criteria. In Figure 4 it is possible to see a generalized scheme of supervised learning. We have a training set, the set of labelled examples, that is subjected to the algorithm, which observing the samples is able to extract a model that can later be used to classify samples without a label.

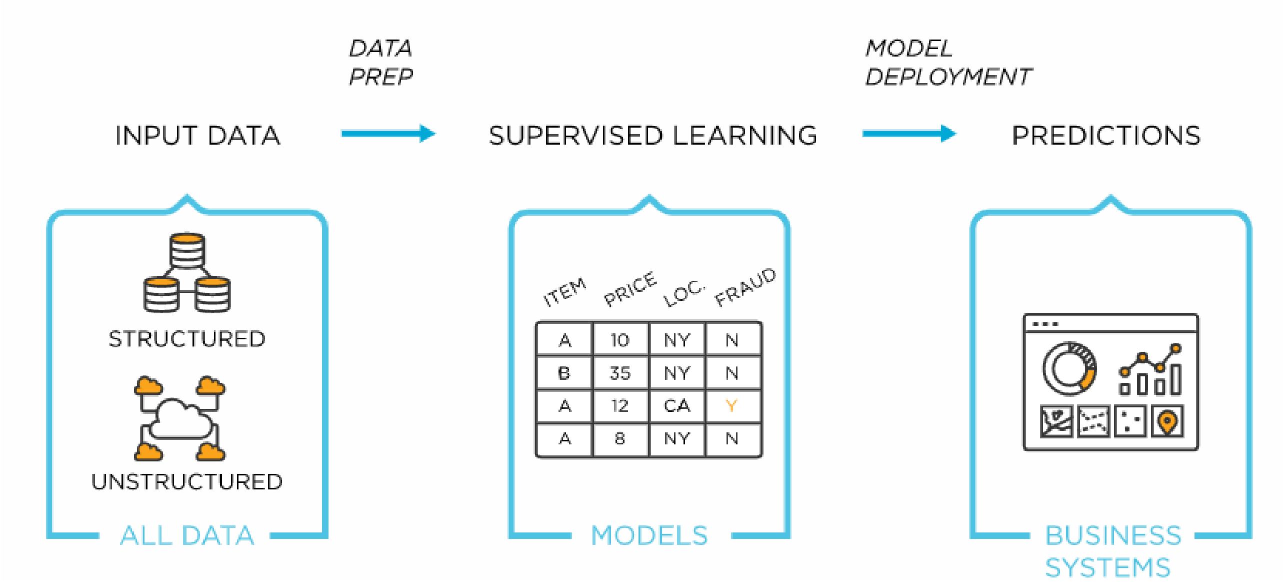


Figure 4: Supervised Learning (from tibco.com)

In this category of algorithm, we could distinguish between two main techniques: classification if the output obtained is a class, a category as in the example of the dog and the cat, or regression if the output obtained is a real number as could be the example of an algorithm that predicts the selling price of a house. Some examples of supervised learning are binary classifiers, multi-class classifiers, linear regression and logistics, and support vector machines (SVMs).

Unsupervised learning is able to work and learn from unlabelled data, which avoids the enormous human effort required to organize and assign labels to all data such as supervised learning. This technique is able to examine the data, identify some common characteristics and divide them on the basis of these characteristics. Doing again the example of dogs and cats, in this case we will be faced with a set of images of dogs and cats not labelled, this time the algorithm will be able to tell us if two images of a dog are similar to each other but different from that of a cat. By exploiting some form of similarity, he will be able to group the data even if he does not know which group they represent. The two main types of unsupervised learning are clustering and data dimensionality reduction. Clustering refers to algorithms that analyse a dataset and divide it into groups containing similar elements to each other, while reducing dimensionality indicates techniques to simplify data without losing too much information.

They are very useful during pre-processing of data to reduce its dimensionality before applying other techniques. Suppose you want to predict the value of a house and have the following features: size, number of bathrooms, number of bedrooms, distance from the nearest school and distance from the nearest hospital. If you want to simplify the dataset, you can group similar classes into one more generic class. For example, the first three could fall under the size category of the house and the last two away from the services. These techniques deal with this task by minimizing as much as possible the number of lost information.

Reinforcement learning is a different type of machine learning algorithm, in which no data is used, but you have an environment and a machine. The machine moves inside the environment and through sensors receives a series of feedback from the environment, which can be positive or negative. Initially the movements of the machine will be random and will gradually be modified according to the feedback received, with the tendency to replicate those that have been positively evaluated and to avoid those that have received negative feedback.

The ancestor of machine learning is the Perceptron (Rosenblatt, 1958). First introduced by Frank Rosenblatt in 1958, it is a binary classifier algorithm that uses the supervised learning method. Its structure is inspired by the neurons of the human brain. As can be seen from Figure 5, the Perceptron takes as input a vector representing the features that are used to distinguish classes and a weight vector indicating the importance to be assigned to each feature. We think for example of wanting to distinguish between motorcycles and machines and to use as features height, width and colour. If in the images we use for his training we have both a motorcycle and a red car, the colour will be associated with a very low weight because it is not recognized as a discriminating feature for the purpose of choosing between bike and car. Inside the Perceptron makes the sum of the products between features and relative weights. The result is then passed through an activation function, in the simplest case the step function, which if the result is greater than zero assigns it to class 1, otherwise to class -1. One of the biggest limitations of this system is that it can only recognize functions that are linearly separable.

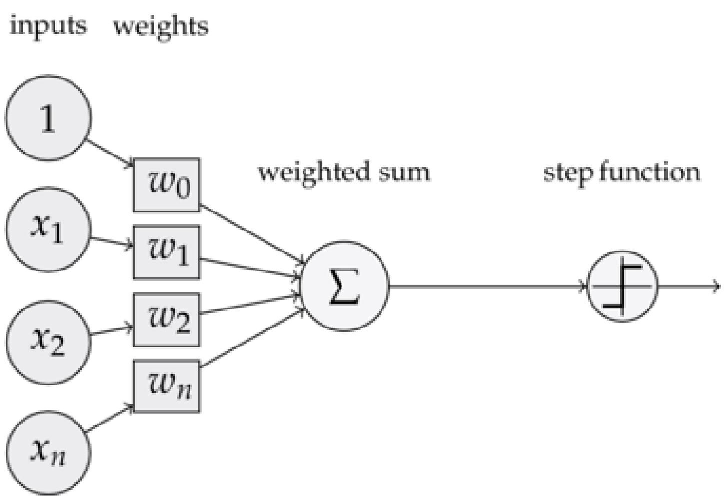


Figure 5: Perceptron structure (from: pyimageserach.com)

2.3.1.1 Deep Learning and Artificial Neural Network

Continuing to be inspired by the structure of the human brain, after the Perceptron that was inspired by the single neuron, neural networks were introduced. A neural network is a structure composed of several interconnected levels, each of which is formed by a number of artificial neurons (Perceptron), as can be seen from Figure 6. The simplest model that can be considered is the one formed by only two layers, one of input and one of output, but usually more complex models are used, with further intermediate layers called hidden layers. The more complex the problem is, the more layers are needed. When we consider multi-level networks, we talk about Deep Learning.

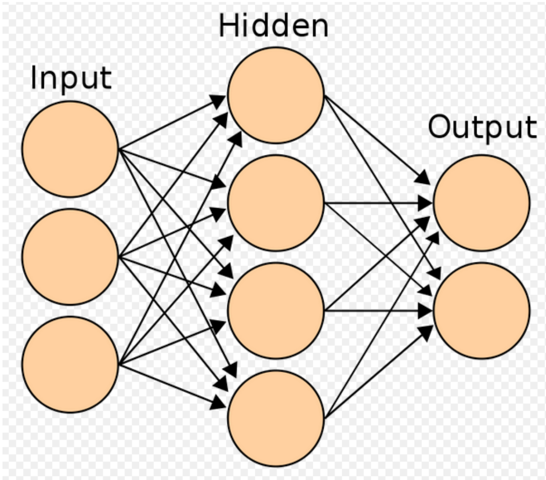


Figure 6: Neural network structure (from Wikipedia)

In order to function at its best, this type of technology needs large amounts of data to analyse and a large number of computational resources. In recent years the availability of these two elements has significantly increased and consequently also the use and development of neural networks.

Artificial neural networks have some characteristics that are interesting in many research fields and application domains. While many of these features vary from model to model, there are some that are sufficiently general:

Robustness - a neural network is able to continue to provide a correct answer even if some of its connections are removed (or " *damaged* ") or if noise is added to the input signal, transmission channels or activation function of neurons. This characteristic is also common to biological nervous systems where the ability to learn and remember is not substantially altered by the continuous loss of neurons. Also, as in the case of biological nervous systems, injured artificial neural networks can sometimes be "*retrained*" to acquire lost skills. These properties represent an advantage over the way serial systems operate where usually the loss of a single link in the processing chain leads to a catastrophic drop in the performance of the entire system.

Flexibility - a neural model can be used for many different purposes: it does not need to know the properties of the specific application domain because it learns them from experience. This does not mean that any one neural model can be used for all kinds of tasks, but it does imply that the user does not necessarily have to know the detailed and analytical solutions that characterize the problem under investigation. In general, the user of a neural network must be able to precisely identify the aims of the project, the type of task and a series of constraints in order to evaluate which neural model is most appropriate. Further knowledge on the nature of the problem can be used to decide other important aspects of the neural model such as the architecture, the type of encoding of external stimuli and the network response, the training parameters, etc. advantages because it allows to face many problems whose analytical solutions are not known. Generalization - a neural network that has been trained on a limited number of examples is capable of producing an adequate response to input patterns that are not used in learning. This property stems in part from the fact that many neural models intimately represent a number of stimulus-response associations greater than the number of synapses available. The neural network tends to extract the invariant characteristics of the input patterns rather than memorizing each individual pattern. The extent to which a neural network is able to generalize to a new input pattern depends on the degree to which these invariant characteristics can be found in the pattern in question. The extraction of invariant characteristics turns out to be a very powerful and economical calculation strategy (which is widely used by biological nervous systems) and the ability to generalize to new stimuli is a very appreciated feature in the typical fields of application of neural networks where it is often impossible to obtain an exhaustive collection of all the data on which the neural network will have to operate. Content-based recovery - artificial neural networks are able to recover their content-based memories from incomplete, similar, or noise-corrupted data. As in biological nervous systems, a clue is sufficient to direct the activation of the system in the appropriate direction by completing and recovering the entire memory. Serial computers, on the other hand, retrieve data by recalling a number that represents the corresponding memory address; if this number is altered or lost, it is no longer possible to recover the entire data.

The choice of a particular learning procedure depends on the objectives that are required to be achieved by the network. The purposes for which ANNs are most frequently used are two:

- Approximation - Assuming that we look for the functional relationship $y = f(x)$ that binds the input vectors with the respective desired output vectors. The neural network is required to approximate the function $f(\cdot)$ through the use of p pairs of exemplary vectors

$\{(x_1, y_1 \ldots., x_p, y_p)\}$. The approximation can be effectively solved by a supervised neural network.

- Pattern classification - in this type of learning the objective is to classify all the inputs provided in a fixed number of categories. To solve this problem the neural network is repeatedly presented in input a series of example vectors together with the indication of the category to which each of them belongs. At the end of the learning process, a "new" vector will be supplied as input, but which belongs to one of the categories that have been stored, with the aim of classifying it correctly. Such a task can be effectively performed by a supervised neural network and associative memory.

The classic application cases are the recognition of text, images and voice but also video sequence as in the case that I will analyse later.

Different neural network structures are available, depending on the purpose for which they are used:

- Forward feed networks (Kayzoglu, 2001) can conduct information in only one processing direction. They don't have cycles and they don't have memory of input happened to times precedence.

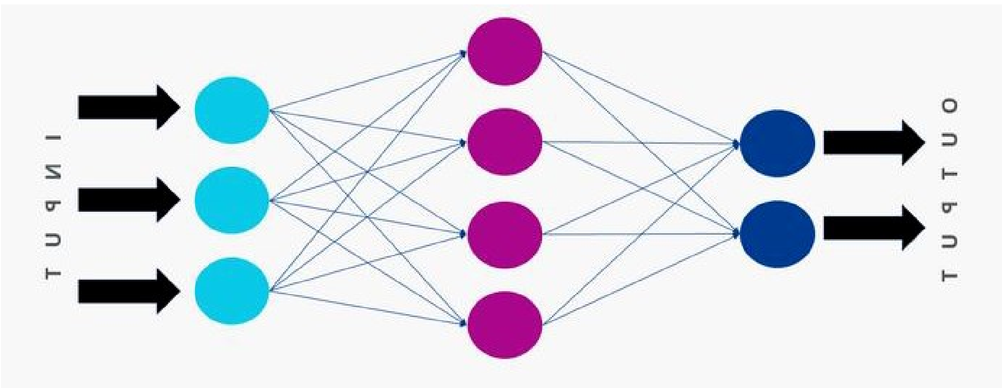


Figure 7: Forward feed network (from ionos.it)

- Recurring networks (RNN) (Rumelhart, 1986), the output values of a layer of an upper layer are used as input to a lower layer. This type of structure allows the system to create a memory. There are several types of recurring networks, among the most significant are the Long Short-Term Memory (LSTM), which are able to learn from long time sequences and preserve the memory.

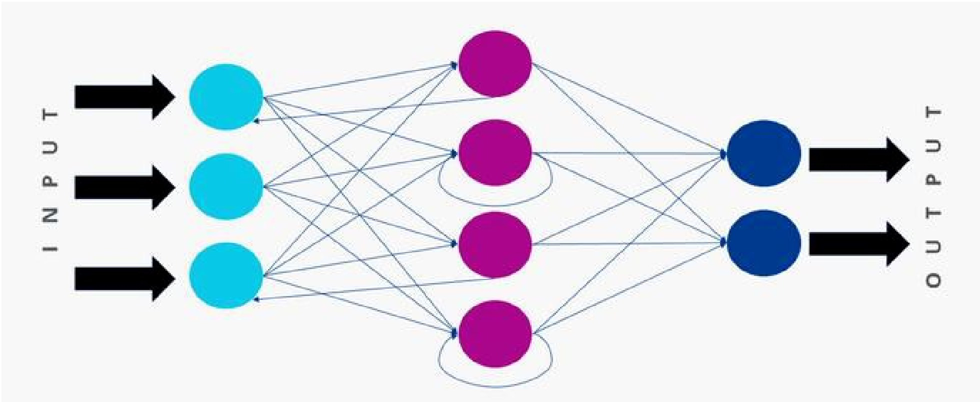


Figure 8: Recurring network (from ionos.it)

- Convolutional networks (LeCun et al., 1989), are feed forward networks, composed of an input level, an output level and some hidden layers that include particular levels such as convolutional, activation or pooling. They are widely used in the field of computer vision and represent the state of the art for many applications such as image classification. It is also the type of network that is used in this study and for this reason a more complete explanation will be dedicated to it in the next chapter.

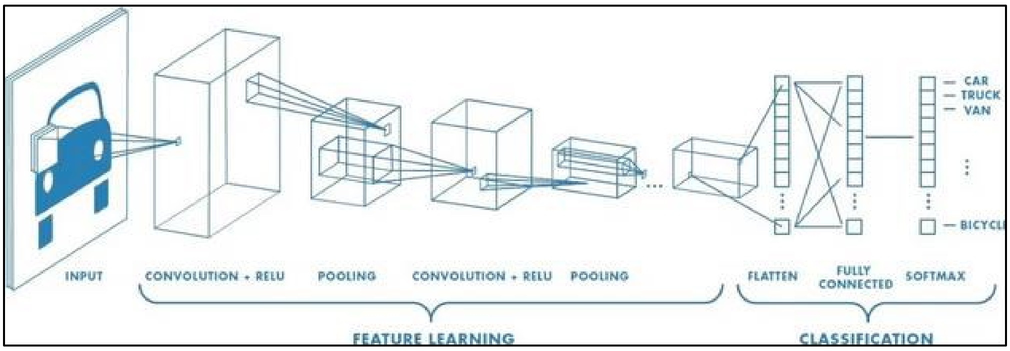


Figure 9: Convolutional networks (from Matworks.com)

2.3.1.2 Convolutional Network

Convolutional networks are typically very deep, because each layer that composes them is only able to recognize a specific characteristic of the image. The first levels deal with the lowest level features such as angles and lines, while continuing with the levels you get to distinguish whole parts of the image such as wheels or parts of the human body.

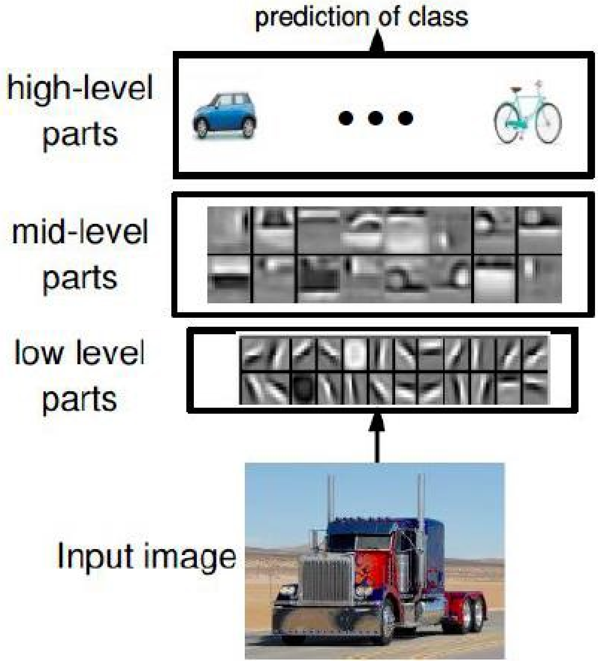


Figure 10: Example extraction features

Convolutional networks mainly perform four operations: convolutions using filters, non-linearity applications, pooling and normalization. For this to happen correctly, there must be a sequence of layers of different types: convolutional layer, pooling layer, batch Normalization layer and fully connected layer.

Convolutional Layer

A convolutional layer applies small filters to a small portion of the image. This operation is iterated to cover the entire surface of the image. A filter is called this because it filters the image based on specific features. For example, a corner filter will be able to locate all the corners within the image. One of the most used sizes for filters is 3x3. Each filter is scrolled over the image to analyse it all. There is a parameter called stride, which indicates how much the filter should be moved before being applied again. The result of applying a filter will be a number, all these numbers put together will make up an output array called a feature map. In Figure 11 we can see how a 3x3 filter is applied.

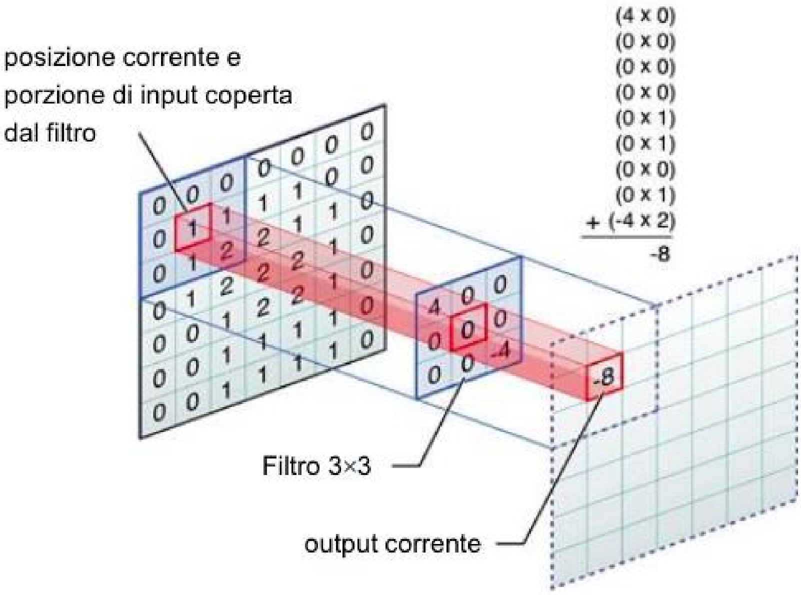


Figure 11: Filter Application (from domsoria.com)

In reality, not only a filter is applied, but more filters are applied to each convolutional level in order to distinguish more characteristics. Not only 2D filters are available, but we can also use multiple filters if necessary.

To apply a filter means to carry out a convolution, that is an operation between two matrices that have the same dimensions, in particular it is the sum of the multiplications between each pair of cells having the same indices. This operation is then applied between the filter matrix and the matrix that represents the piece of image that is considered at that time.

Pooling layer

The pooling layer is a method of reducing resolution and making the representation smaller and more manageable. Unlike convolution that acts on the whole set of feature maps, this type of layer operates on each of them. When pooling, a square matrix is used which is scrolled over the input data as is done for filters. Also, in this case the stride is used to indicate how much to move the matrix.

There are two main types of pooling:

- Max pooling is the easiest and most used. Returns the maximum value present within the considered window. The idea behind it is to keep the information relevant and discard the less significant.

- Average pooling, to reduce the number of discarded information compared to the use of max pooling, is performed the average of the values in the window considered. In this way the output depends on all the input and not only on a value of it.

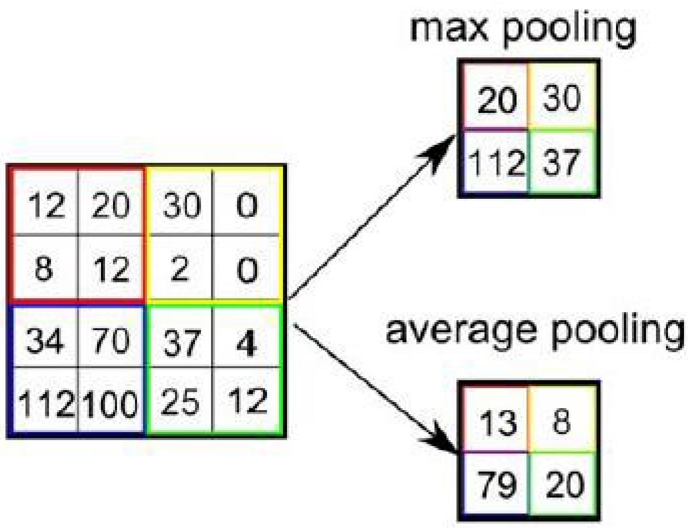


Figure 12: Pooling examples (from datascience.eu)

Fully connected layer

At the end of a convolutional network, a fully connected layer is always placed after the last level of maxpooling. It owes its name to the fact that every neuron that composes it, receives in input all the values deriving from the previous level, taking up the structure of the perceptron multilayer. The goal of this level is to combine all the data received in input to assign the label to the image.

Batch Normalization Layer

Convolutional neural networks are characterized by the problem of normalization. Almost all machine learning algorithms work best with normalized data. Although the input data is normalized, it is no longer normalized after each convolutional or fully connected level. This problem slows down the training time of the network and worsens its performance. The solution is to normalize the data after each convolutional/fully connected layer. This operation is handled by batch Normalization layers.

2.3.1.3 The Activation Functions

The activation function Φ represents the "rule" according to which the neuron provides the response to a given net input A received at the input. In practice, this function must “leave the channel open” if A is above a certain threshold and close it if it is not. Two classes of activation functions can be identified:

- functions without saturation, that is, not limited either above or below, capable at most of effecting a " smoothing " on the values that are treated by them;
- functions with saturation, ie with a domain included within certain values and therefore capable of limiting the output values of neurons within particular ranges, introducing an “internal” non-linearity to the functioning of the network.

The most used activation functions are presented below.

✓ Continuous linear function *(without saturation)*:

$$\Phi_{(x)} = kx \text{k = constant} \quad (1)$$

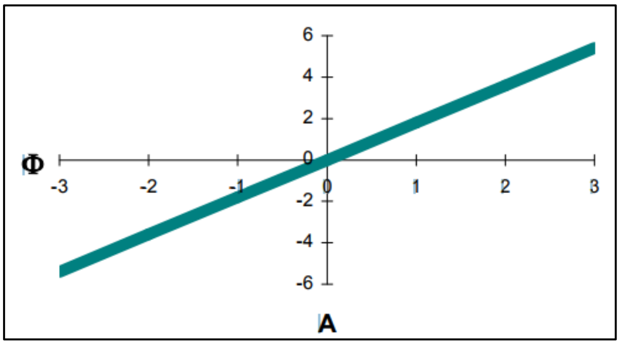


Figure 13: Linear function

This type of function allows the neuron to transmit a gradation of signals of varying intensity that can be suitably exploited by the receiving neurons: this property is considered analogous to the firing frequency (impulses per unit of time) of biological neurons.

✓ Sigmoid function *(with saturation)*:

$$\Phi_x = \frac{1}{1+ e^{-kx}} \text{k = constant} \quad (2)$$

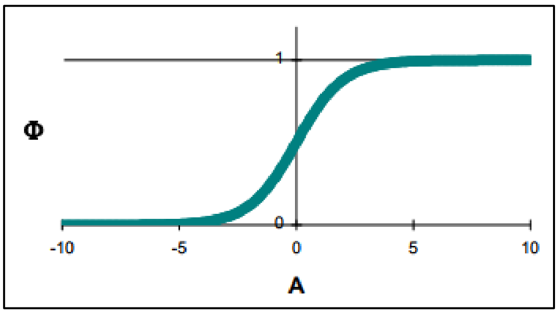


Figure 14: Sigmoid function

It is an asymptotic function towards 0 and 1, belonging to the class of continuous nonlinear functions. The constant k controls the inclination of the central part. For k →∞ the sigmoid function approximates the *step function*.

- ✓ Arctangent function *(with saturation)*:

$$\Phi_{(x)} = \arctan x \tag{3}$$

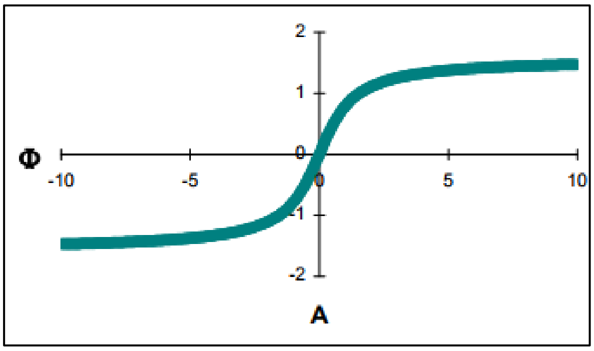


Figure 15: Arctangent function

This function (which is also non-linear) has codomain [- π/ 2, π/ 2] so that, unlike the sigmoid, it can also assume negative values.

In principle, it is possible to associate different activation functions from neuron to neuron. However, in most models, an identical activation function is used for all neurons (or at least for all neurons in the same layer). The neurons of the entrance layer are an exception, to which a linear activation function is always associated.

Feedforward Neural Networks and Multilayer Perceptrons

We refer to the simple three-layer feedforward network represented in Figure 16. The hidden layer is connected to the input and output layers by synapses characterized by modifiable weights. There is also a single unit of bias connected to each node of the hidden layer and the output layer. The input vector x = (x_1... x_2, x_d) is "submitted" to the input layer, and the output of the i-th input node equals the component $x_i$of this vector since such neurons are assigned a linear activation function with k = 1.

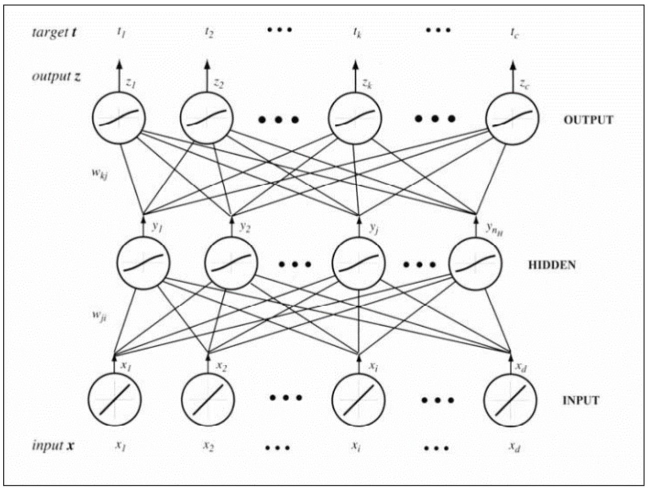


Figure 16: Scheme of a three-layer feedforward network (from cedar.buffalo.edu)

It is also assumed that the neurons of the input layer and those of the output layer are characterized by the same non-linear activation function $\Phi_{(.)}$.

The network computation process takes place according to the following steps:

1) Each neuron in the hidden layer calculates its own net input:

$$A_j = \sum_{i=1}^{d} w_{ij}\, x_i + w_0 = \sum_{i=0}^{d} w_{ji}\, x_i = \boldsymbol{w_j x} \qquad j = 1, \ldots .\, n_H \qquad (4)$$

where $\boldsymbol{w_j} = w_{j1}, w_{j2}, \ldots ., w_{jnH}$ is the vector of the weights associated with the synapses of the *j-th* hidden neuron.

2) Each neuron of the hidden layer outputs the value of the activation function at its net input:

$$y_j = \Phi\left(A_j\right) \qquad j = 1, \ldots \ldots ., n_H \qquad (5)$$

3) Similarly, each output unit calculates its net input on the signals received from the hidden unit as:

$$A_k = \sum_{i=1}^{nH} w_{kj}\, y_j + w_{k0} = \sum_{i=0}^{nH} w_{kj}\, y_j = \boldsymbol{w_k y} \qquad k = 1, \ldots .\, c \qquad (6)$$

4) Finally, each neuron of the output layer calculates the value of the activation function at its net input:

$$z_k = \Phi\left(A_k\right) \qquad k = 1, \ldots \ldots, c \qquad (7)$$

This last value represents the *k-th component* of the output vector
z = ($z_1, z_2, \ldots \ldots ., z_k$*)* of the network.

A particular neural network structure is called Multi-Layer Perceptron (MLP) and is a multilayer Feed Forward network consisting of a single artificial neuron, whose activation function is the sgn () function. This type of elementary neural network allows to implement only a few functions, such as logical AND or XOR.

However, by complicating the network structure with multiple layers, and by modifying the activation functions of artificial neurons, it is possible to train the network to approximate any function and solve a wide range of problems, as the universal approximation theorem tells us.

The MLP network is the natural extension of the perceptron introduced in the Figure 17. Three blocks are identified in it:

1. Input units: the number of neurons is equal to the size of the input vectors.
2. Hidden layers: are layers of hidden neurons in which the processing of inputs takes place; as will be widely seen in the section dedicated to Deep Learning, these are the layers in which the network stores its representation of the data it receives. The hidden layers can be one or more. Their number defines the depth and complexity of a neural network.
3. Output unit: the number of neurons, as in the input section, is equal to the size of the output space.

It is important to underline that MLP is a fully network connected: each neuron of a layer is necessarily connected to at least one neuron of the next layer (Dolgui et al., 2021).

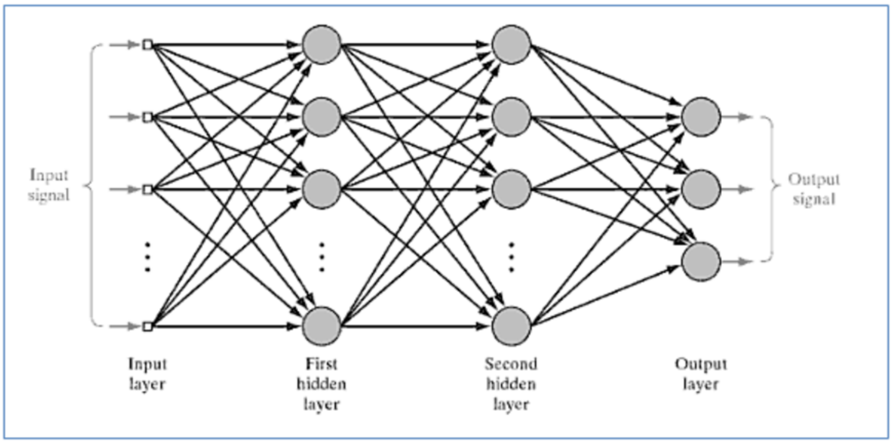


Figure 17: Architecture of an MLP

The expressive power of a neural network refers to the number of functions it is able to implement. It is clear that a non-linear multilayer network has a higher expressive power than a network without hidden layers. Kolmogorov has shown that any continuous function can be interpolated by means of a three-layer network having an adequate number of hidden units n_H, suitable weights and suitable activation functions. In particular, Kolmogorov proved that any continuous function *g (x)* defined in a hypercube with unit sideI^n *(I = [0,1], n ≥2)* can be represented in the form:

$$g(x) = \sum_{j=1}^{2n+1} \xi_j \left(\sum_{i=1}^{d} \psi_{ij}(x_i)\right) \quad (8)$$

where ξ_j and ψ_{ij} are functions chosen in an opportune way. The equation can be expressed through a three-layered neural network (having *d* input neurons and *2n + 1* hidden neurons) as follows:

a) Each of the *2n + 1* hidden units receives as input the sum of *d* non-linear functions, one for each x_i input value.

b) Each hidden unit has a non-linear function ξ of its activation as output.

c) The output unit simply adds up the contributions of the hidden units.

Unfortunately, it is very difficult to define a neural network that perfectly interpolates a function *g (x)* starting from *Kolmogorov 's theorem* for many reasons. The fundamental reason lies in the fact that the functions ξ_j and ψ_{ij} are very complex and have an extremely irregular trend, while the regularity of the trend of an activation function is very important for learning through the descending gradient rule. On the other hand, *Kolmogorov 's theorem* says very little about the pivotal problem in the question of interpolation based on artificial neural networks: how to define nonlinear functions starting from the knowledge of data. A more intuitive proof of the expressive capacity of a three-layered neural network is provided by the well-known *Fourier theorem* according to which a continuous function *g (x)* can be approximated arbitrarily accurately through an infinite sum of harmonic functions. It is possible to imagine a network whose hidden units implement harmonic functions by equating the weights of the connections between the hidden layer and the output layer to the Fourier coefficients.

2.3.1.4 The Backpropagation Algorithm

As shown in the previous paragraphs, based on the learning mode, ANNs are divided into two classes: supervised neural networks and unsupervised neural networks.

The Backpropagation algorithm is one of the simplest and most general methods for supervised learning of multilayer neural networks. It was first proposed by Rumelhart, Williams and Hilton in 1986 and essentially operates by modifying the network parameters in order to minimize the error between the output of the output units and the target values corresponding to the input vector presented at the net. The *training error* (or *criterion function*) is a scalar function of the weights and tends to zero when the weights assume values such as to make the outputs supplied by the network equal to those of the target. Formally, this error is defined as the sum, extended to all output units, of the difference between the desired output t_k and the real output z_k, squared:

$$J_{(\widetilde{w})} = \frac{1}{2}\sum_{k=1}^{c}(t_k - z_k)^2 = \frac{1}{2}\|t - z\|^2 \qquad (9)$$

where *t = (* $t_1, t_2, \ldots, t_c$) and $z = z_1, z_2, \ldots, z_c$) are respectively the vectors of:

target and output from the network while $\widetilde{w} = w_{pq}$it is the matrix of the network weights. The backpropagation learning rule is based on the descending gradient method. The weights are initialized randomly and are changed in the direction in which the error decreases:

$$\Delta\widetilde{w} = -\eta \frac{\partial J}{\partial \widetilde{w}} \quad (10)$$

or in components:

$$\Delta w_{pq} = -\eta \frac{\partial J}{\partial w_{pq}} \quad (11)$$

where ηa parameter that takes the name of learning rate and indicates the amplitude of the variation of the weights The algorithm requires that the matrix of weights be considered iteratively and modified as follows.

$$\widetilde{w}(m+1) = \widetilde{w}(m) + \Delta\widetilde{w} \quad (12)$$

where *m* indicates the particular pattern presented.

Equation (12) is solved for a three-layer network. With reference to Figure 14, we initially consider the weights between the units of the hidden layer and those of the output layer $\{w_{ij}\}$. Since the error does not explicitly depend on $\{w_{ik}\}$, we can use the derivation rule of compound functions:

$$\frac{\partial J}{\partial w_{kj}} = \frac{\partial J}{\partial A_k}\frac{\partial A_k}{\partial w_{kj}} = -\delta_k \frac{\delta A_k}{\delta w_{kj}} \quad (13)$$

where $\delta_k$is the sensitivity of an output unit defined as:

$$\delta_k = -\frac{\partial J}{\partial A_k} \quad (14)$$

This quantity indicates the variation of the overall error as a function of the net input of the unit considered. If the activation function $\Phi_{(.)}$is differentiable, then differentiating the equation (14) we obtain:

$$\delta_k = -\frac{\partial J}{\partial A_k} = -\frac{\partial j}{\partial z_k}\frac{\partial z_k}{\partial A_k} = (t_k - z_k)\Phi'(A_k) \quad (15)$$

We can calculate the last derivative in equation (15) using (13):

$$\frac{\partial A_k}{\partial w_{kj}} = y_j \quad (16)$$

By combining these two results, we obtain the learning rule according to which to modify the weights that connect the hidden layer to the output one:

$$\Delta w_{kj} = \eta\, \delta_k y_i = \eta\, (t_k - z_k)\, \phi'(\mathrm{A}_k)\, y_j \quad (17)$$

Let us now consider the learning rule of the weights that connect the input state to the hidden one. Using the differentiation rule of compound functions again, we get:

$$\frac{\partial J}{\partial w_{ji}} = \frac{\partial J}{\partial y_j}\frac{\partial y_j}{\partial \mathrm{A}_j}\frac{\partial A_j}{\partial w_{ji}} \quad (18)$$

The first term to the right of (44) can be developed as follows:

$$\frac{\partial J}{\partial y_j} = \frac{\partial}{\partial y_j}\left[\frac{1}{2}\sum_{k=1}^{c}(t_k - z_k)^2\right]$$

$$= -\sum_{k=1}^{c}(t_k - z_k)\,\frac{\partial z_k}{\partial y_j}$$

$$= \sum_{k=1}^{c}(t_k - z_k)\,\frac{\partial z_k}{\partial \mathrm{A}_k}\,\frac{\partial \mathrm{A}_k}{\partial y_j}$$

$$= \; -\textstyle\sum_{k=1}^{c}(t_k - z_k)^2\phi'(\mathrm{A}_k)\,w_{kj} \qquad (19)$$

The final sum over the output units in eq. (19) indicates how the outputs of the hidden units $y_j$affect the error of the output units. In analogy with what has been done for the output units, we can define the sensitivity of a hidden unit using eq. (19):

$$\delta_j = \; \phi'(\mathrm{A}_k)\;\textstyle\sum_{k=1}^{c}\left(w_{kj}\,\delta_k\right) \qquad (20)$$

The sensitivity of a hidden layer unit is simply the sum of the individual sensitivities of the weighted output units w_{kj}, multiplied by $\Phi'(A_j)$.

The learning rule for the weights that connect the input layer to the hidden one is, then, the following:

$$\Delta w_{kj} = \eta x_i\,\delta_j = \eta\; x_i\phi'(\mathrm{A}_k)\;\textstyle\sum_{k=1}^{c}\left(w_{kj}\,\delta_k\right) \qquad (21)$$

The learning rules (17) and (21), used in an appropriate training protocol, provide the *Backpropagation algorithm* or " *backward propagation (of the error)* ", so called because during the training the error must be propagated from the output layer to the hidden layer so that learning of the weights that connect the input layer to the hidden layer takes place. Ultimately, the "*backpropagation*" is nothing more than the descending gradient method applied to stratified models, in which the application of the derivation rule of compound functions allows us to calculate the derivatives of the criterion function with respect to all the weights of the model. As with all methods based on gradient-descending procedures, the behaviour of the *Backpropagation algorithm* depends on the initial conditions. It would then seem natural to initialize all weights to zero, but equation (21) shows that such an approach could have undesirable consequences. In fact, if all the initial weights w_{kj}were null, the backward propagated error would in turn be null and the weights that connect the input layer to the hidden one, w_{ji}would never change. For this reason, the weights are initialized randomly.

Supervised learning consists in presenting to the network those patterns (the training set) to which we know to match the precise output values, (the target values), determine the network output and change the weights to match the actual network output to the target values.
The most important training protocols are (Dolgui et al, 2021):

- *Stochastic protocol:* According to this protocol, the patterns are chosen randomly from the *training set*, and the network weights are changed every time a pattern is presented. This method is called stochastic because the training data can be considered as random variables.
- *Bach protocol:* In batch training, the patterns are not randomly selected, but are presented all together to the network. The examples are provided to the network for a certain number of epochs (where an epoch corresponds to single presentations of all the patterns of the training set) and after each epoch the weights are modified. This process is iterated until a properly defined *stop condition occurs.*
- *On-line training:* According to this protocol, each pattern is presented once and only once; there is no use of memory for storing patterns.

2.3.1.5 Generalization and overfitting

The generalization consists in producing an appropriate response for a new input pattern that was not included in the training group without further modifying the synaptic connections. The problem of an adequate generalization arises when we only have a limited number of training patterns but want to ensure good network performance at the end of the learning process. A large number of resources (weights and units) allows the network to learn a wide range of specific functions that are responsible for the exact correspondence between inputs and outputs for training patterns, thus decreasing the likelihood that the network will discover the very general function that describes the whole domain of the problem and that would result in correct answers for the test patterns as well. Too large a number of connections also compromises convergence towards a global minimum, both because it increases the complexity of the error surface and because it requires a large number of training patterns. One way to limit the problem of poor generalization without intervening on the architecture of the same, is to stop the training before the network learns "too well" the relationship at stake. In this way we try to reproduce a situation in which the network, although it does not reach an overall minimum for the training patterns, is nevertheless able to obtain an acceptable level of performance both on the training patterns and on the new patterns that are presented to it. There are also simple empirical strategies to help you make this decision when choosing the exact moment to stop training. The most used method consists in dividing the available patterns into two parts (not necessarily the same): one part is used for learning (*training patterns*), and another to evaluate generalization skills (*test patterns*). Assuming that both sets of patterns are uniformly extracted from the same distribution of data, the error on the test patterns initially drops similarly to the error on the training patterns, but at some point, it reverses direction and begins to grow while the error on training patterns continues to decrease as seen in Figure 18. This trend reversal indicates the start of training data *overfitting* and is a good indicator of when to stop training. However, when the number of initially available patterns is very scarce or not very redundant, this method is not applicable and, in this case, it is advisable to stop training as soon as the curve that describes the fall of the error on the training patterns slows down considerably its descent.

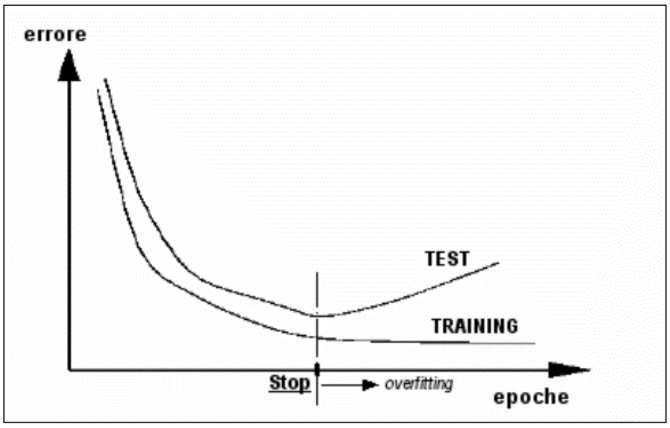


Figure 18: Overfitting

2.3.2 Fuzzy Logic

Fuzzy Logic is an important part of soft computing which imitates the notable capacity of the human mind to reason and learn within an environment made of qualitative evaluations (Savino et al, 2017).

The fuzzy logic is based on the so-called fuzzy sets, or an extension of the traditional concept of a set, called crisp sets, to distinguish them from the previous ones.

A set is composed of all the elements of the universe of discourse that satisfy a given membership function. For a crisp set, the membership function is Boolean, that is, it associates a "true" or "false" value to each element of the discourse universe depending on whether the element belongs to the whole or not. There are cases, however, in which the belonging or not of an element to a certain set may not be well defined, i.e. the need arises to define how much an element of the

universe of discourse possesses a certain property, or, in other words, how much it can belong to the set of elements that possess that property. It is therefore possible to define a membership function that returns a value between 0 ("false") and 1 ("true"). This allows us to specify how much it is believed that an element of the universe of discourse belongs to the whole, that is, it allows to give a degree of belonging to the whole that is not necessarily Boolean (Hájek, 2013).
The definition of a fuzzy set is only an extension of the classic definition of a set. It has a border that is no longer a clear line of demarcation between the elements that belong to the whole and those that do not belong to it, but an area in which there are elements that can be classified as belonging to the whole with a certain degree. For this reason, the whole is called "fuzzy" (Trillas and Eciolaza, 2015).
Figure 19 shows the typical structure of a fuzzy system.

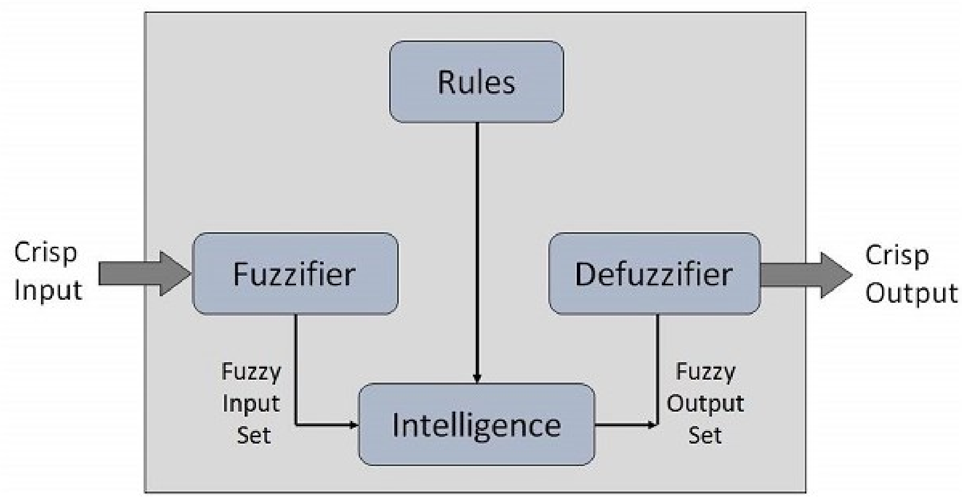


Figure 19: Structure of a fuzzy system (Satam, 2022)

It has four main parts as shown:

- *Fuzzification interface*

The fuzzification interface has the task of measuring the input variables to the controller and converting the numerical values of the input data into appropriate linguistic values, to which fuzzy sets correspond. In control applications a singleton fuzzification is usually used: the measured numerical value e_0 corresponds to a fuzzy value with unit membership function at the point $e = e_0$ and null for all the other values of the variable. In applications where the input signals are notably affected by measurement noise, it may however happen that the fuzzification is carried out by means of triangular membership functions centered on the measured value, which lead to a more general and therefore more complex inference mechanism than to the previous case.

Another task of the interface that performs the fuzzification is the choice of a conversion scale to switch from the intervals in which the input variables range to the corresponding universes of discourse (normalization). This transformation is affected through the gains placed in input to the controller, whose calibration constitutes one of the crucial problems of the design of a fuzzy controller; a similar gain is present at the controller output, at the defuzzification interface output.

- *The knowledge base*

The knowledge base of a fuzzy controller contains the set of rules based on which the control variables must act on the system to impose the desired behaviour on it. The fuzzy rules are linguistic and are expressed based on the opinions of experts or specialized operators. To define the linguistic algorithms that constitute the rules of the fuzzy system and that synthesize the knowledge of the process, the experts carry out a subdivision of the input universes into fuzzy sets that identify the representative labels for the input and output variables of the system. Note that the fuzzy number of labels a universe divides into (cardinality) is of fundamental importance, as it affects the sensitivity of the system and the maximum number of rules allowed. An appropriate compromise must therefore be sought in defining this parameter. In general, in applications where the variables have a symmetrical universe with respect to the origin of the axes, an odd cardinality is considered. In addition to the number of fuzzy labels associated with a linguistic variable, the type of membership function that describes them is crucial. Figure 20 shows some of the most common membership functions used. Generally, that the linear ones (triangular, trapezoidal, "S" or "Z") are among the most common, as they can be easily implemented.

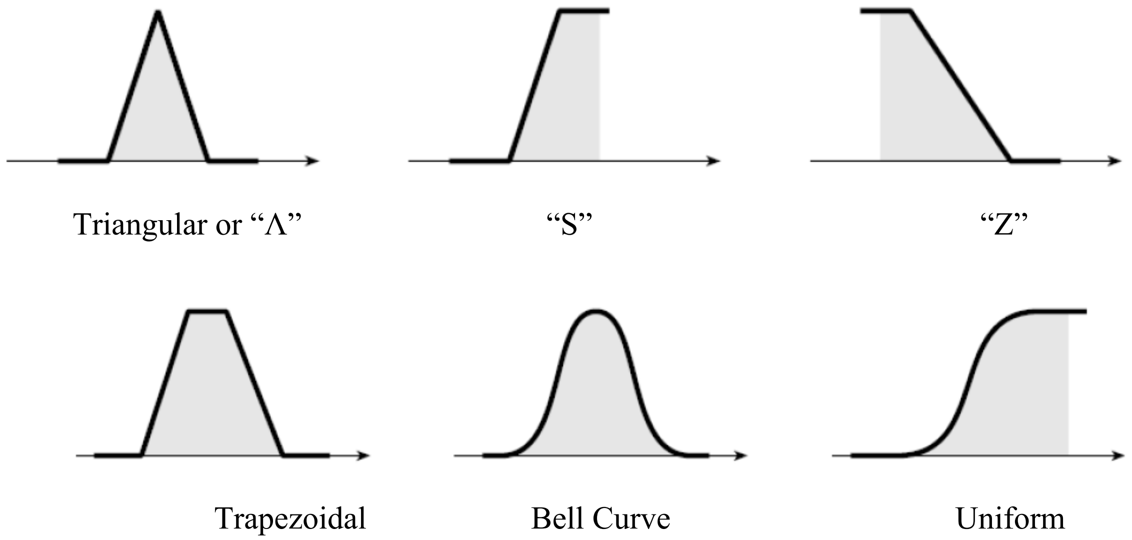


Figure 20: Trends of the most common membership functions

It is also important that the membership functions of the labels of a variable overlap in an appropriate but not excessive way, so that for each value at least two rules with an average degree of truth are defined. Therefore, the level of completeness or completeness is defined e, defined as the maximum ordinate ε for which the ε-cuts of all labels cover the entire universe. Completeness describes the aptitude of the fuzzy control algorithm to infer a control action with confidence not less than ε. Based on heuristic considerations, a completeness = 0.5 is suggested in the literature, so that for each input value there is always a dominant rule, that is, with a degree of truth greater than or equal to 0.5. In Figure 21 three different cases of completeness are represented, corresponding respectively to e = 0.5, e = 0.25 and e = 0.7.

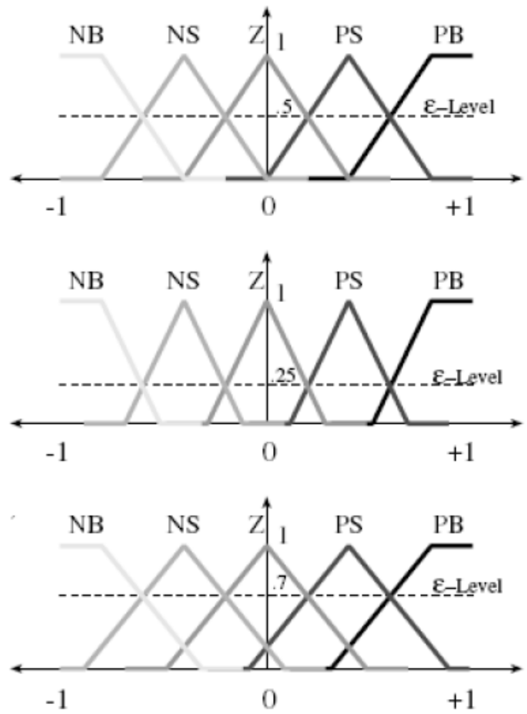


Figure 21: Examples of different levels of completeness

- *The inference engine*

The inference engine or decision logic of the fuzzy system processes the rules based on the conditions of the system in the instant of time considered, through the inference procedure.

The three main design choices to be made to define the operating mechanism of the decision logic are the type of conjunction operator (and), the type of aggregation operator (or) and the type of implication or composition operator (if ... then, →).

For each rule, the truth value of the antecedent is calculated by joining the degrees of belonging of the labels into the input variables; then the truth value of the consequent is calculated by applying the composition rule; finally, the truth values of the consequents are aggregated, determining the fuzzy set of the conclusion.

- *The defuzzification interface*

The defuzzification interface converts the conclusions of the inference process from the linguistic form, which is expressed in terms of the membership function of a fuzzy set, into numerical form. Also, in this interface can be present the normalization gains, which allow to pass from the speech universes of the controller output variables to the corresponding intervals of the numeric variables. The defuzzification methods present in the literature of the sector are many. The two most common are:

- Center of gravity (COG) method: the numerical value obtained from defuzzification is the center of gravity of the fuzzy set provided by the decision logic;
- Mean of Maximum (MOM) method: the numerical value obtained from the defuzzification is given by the average of the points for which the membership function of the fuzzy set provided by the inference engine is maximum.

2.4 MAINTENANCE THROUGH ARTIFICIAL INTELLIGENCE

Machine Learning (ML) approach could help the application of PMS as it is a useful tool to analyse data and monitor the system operational model. ML is a selection of different algorithms that aims to analyse and process data for clusterisation, classification and prediction purposes (Hofmann, Schölkopf, and Smola, 2008). The final goal of ML approaches is to look for complex relationships in the data that may be difficult to capture by common tools, to detect incoming failures and to determine higher accuracy in the RUL prediction of systems (Wagner et al., 2016). ML techniques applied in PMS have gained an increasing interest both in academia and industrial communities since they have shown interesting modeling and forecasting capabilities even in very complex and heterogeneous problem domains (Paolanti et.al., 2017) Several authors in the last years have proposed maintenance models using Machine Learning. Rivas et al. (2019) elaborate a model to establish the RUL of a machine through a recurrent neural network; Jimenez et al. (2018) present an approach to optimize the sensors in a condition monitoring system employing ultrasonic waves and to classify some features through Machine Learning and Neural Network. Sangje Cho et al. (2018) provides a hybrid machine learning approach combining unsupervised learning and semi-supervised learning to manage the lack of annotations describing the machine status or maintenance history in the data amount available in modern manufacturing companies. Salmaso et al. (2019) use a DOE step before the usual Big Data Analytics and machine learning modeling phase to reduce the difficulty of finding causal relationships among variables.

All these models, and many others, are fundamentally based on data collection. Big manufacturing companies collect and handle large amounts of data from manufacturing process and from higher levels of hierarchical control. These data may be useful for implementation of predictive maintenance and ML approaches because the data from lower level of hierarchical control are crucial for predictive maintenance. This trend is strongly related to spread of Industry 4.0 technologies into manufacturing, which represents a basis for data analysis of any kind, including for predictive maintenance (Kagermann et al., 2013).

I4T is mostly focused on the manufacturing area also because this is the most directly implicated in the efficiency and sustainability of the industrial processes, including Maintenance. Implementing concepts like Internet of Things and Big Data in manufacturing companies, with large quantities of data from the manufacturing processes, can be further utilized in predictive maintenance or failure prediction. On this research stream

Thanks to certain I4T it is possible to collect and monitor data for the entire of use of a product (Shin and Jun, 2015). Last decade was abundant of these I4T, such as Radio Frequency IDentification (RFID), optoelectronic sensors, Micro-Electro-Mechanical System (MEMS) and wireless telecommunications, Product Embedded Information Devices (PEID), to cite a few. Such advances in information technology have accelerated growth in the maintenance sector, enabling network bandwidth, data collection and retrieval, data analysis and decision support capabilities for large time series of databases, allowing to diagnose the state of degradation of the product (Prajapati, et.al., 2012). Therefore, the use of these information may give the opportunity to improve the efficiency of product maintenance operations, as it is possible to diagnose product status, predict product anomalies and perform proactive maintenance (Kobbacy et.al., 2008). Paolanti et.al. (2015, 2017) define three types of approach based on data, model and hybrid. They also explain that the data-based approach is usually defined as a data mining approach or machine learn. Within this research stream Wang et. al. (2016) and Susto et. al. (2015) uses historical data for learning, in order to evaluate the behaviour of the system. These data are used in a sustainable manner in areas where the availability of data increases, for example in the industrial sector (Paolanti et.al., 2017 and Naspetti et.al., 2016). This approach can be classified in i) Supervised, in which information relating to errors or faults is present in the modeling data set and ii) Non-Supervised, in which information on the logistics and/or the process is available but of which there is no maintenance data. The model-based approach uses an analytical model to figure out the behavior of the system, while the hybrid one combines supervised and unsupervised learning. The possibility of having information about maintenance depends on the choice of the existing maintenance management policy and, whenever possible, supervised solutions are preferred (Druck et.al 2008). However, the solutions based on machine learning techniques seem to be among the most used, as shown by Heng et.al., (2009) and Su et.al., (2006), which carry out a maintenance analysis for the production of semiconductors.

Manufacturing systems efficiency is of course strongly related to the availability of production systems; anyway, another important parameter is the quality of the throughput, as the most used efficiency index in the world, the Overall Equipment Effectiveness OEE index, considers both availability and quality. In order to reduce defective products, predictive maintenance on production systems can help to keep machinery in good conditions to carry out the standard outputs; predictive maintenance should be used jointly with sampling the output to screen out the defective units (Kuo, 2006). Ross (1971) used such an approach and found the optimal maintenance policy to maintain a Markovian deteriorating production machine. Recently, Dreyfus and Kyritsis (2018) proposed an approach combining zero-defect manufacturing, predictive maintenance and scheduling algorithms which deal with high uncertainty to bring a clean solution that significantly improves production capacity. The increasing numbers of connected machines, systems, equipment and goods in the production systems led to the emergence of the so-called so-called Cyber-Physical Production Systems (CPPSs), and ‘smart machining centers’. On this line, Sezer et al. (2018) developed an Industry 4.0 low-cost architecture focused on predictive maintenance. Their approach was able to predict the machining quality through Recursive Partitioning and Regression Tree model technique. Similarly, Changchao et al. (2017) introduce a product quality oriented predictive maintenance strategy for manufacturing systems. In addition to visual inspection, various automated solutions have recently been developed, based on different techniques, such as pattern recognition, empirical and physical modeling, neural networks or fuzzy logic (Wu et al., 2007). Many of these techniques perform well and allow the desired monitoring of the system. However, their implementation is often correlated with technical limits relative to the complexity of the solution and to legal/financial restrictions (Auf der Mauer et.al., 2018; Hashemian and Bean, 2011).

Spendla et al. (2017) described an approach to build a data storage platform integrating maintenance parameters and process quality to support the analysis and decision phases.

2.5 LITERATURE ANALYSIS RESULTS

According to the analysis of the state of art, we may argue that PMS implemented through I4T may be an efficient way to remove the potential failures and guarantee the stable operation of the manufacturing system.

This review, however, highlights how at the moment there is a gap between the use of maintenance data through ML and the real-time control of product quality through a single framework, as represented in Figure 22 where the references in the literature review are classified by domain and the grey are is the gap in which our research fits.

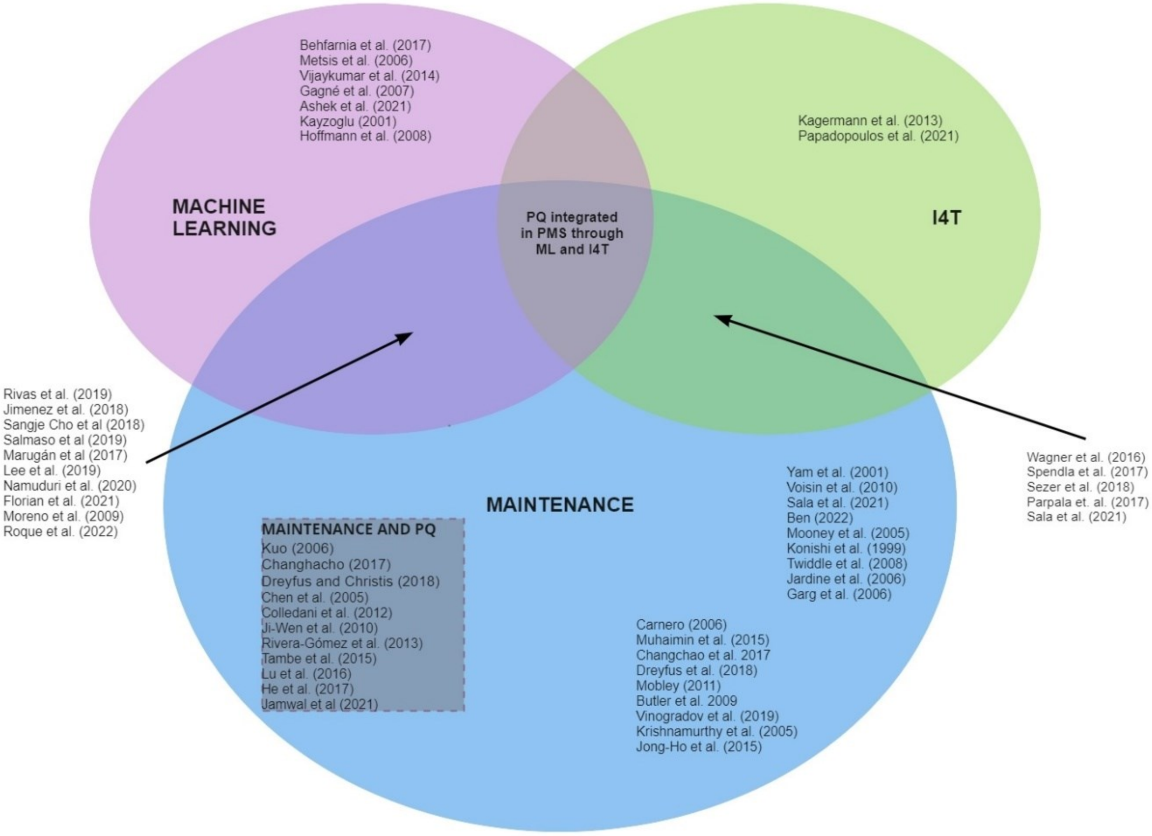


Figure 22: Domains and references in literature review

Furthermore, appropriate PMS approaches may be finalized to improve at the same time the efficiency and reliability of the manufacturing system and the PQ. These approaches should relate certain PQ features to the maintenance status of the machine and through I4T define dynamically the appropriate PMS.

This research is focused to investigate how a model integrating the key PQ variables, and the quality deviation that characterizes the PQ level based on co-effect between manufacturing system component reliability and product quality. The optimal maintenance strategy is achieved by optimizing the quality cost, repair cost, maintenance cost, and interruption cost simultaneously, through a real-time fuzzy agent.

3 RESEARCH DESIGN

3.1 RESEARCH FRAMEWORK

Based on the extant body of literature, this study attempts to front some basic questions regarding the connection between PMS and PQ.

The first research question aims to address the feasibility of a model that considers PQ parameters as potential variables for PMS. The basic investigation is focused on the potential relations of machines' data with predictive maintenance indicators and PQ.

> **RQ$_1$:** How is it possible to consider product quality parameters within the problem of predictive maintenance?

The second research question aims to investigate the potential impact of PMS according to a certain level of PQ. This portion of the research has been conducted by investigating the ability of the model realized to "learn" from the PQ data and to suggest the optimal PMS.

> **RQ$_2$:** May we improve MTBM and decrease maintenance and production costs through the PQ optimization?

The goal of our Research Framework (Figure 23) is to incorporate the Product Quality into the management of Maintenance activities and to implement the suitable PMS according to a variable threshold that describe the quality level the production process must guarantee. The framework can be divided into three main steps, which provides the input variables definition, the predictive analysis model definition and finally validation and utilization.

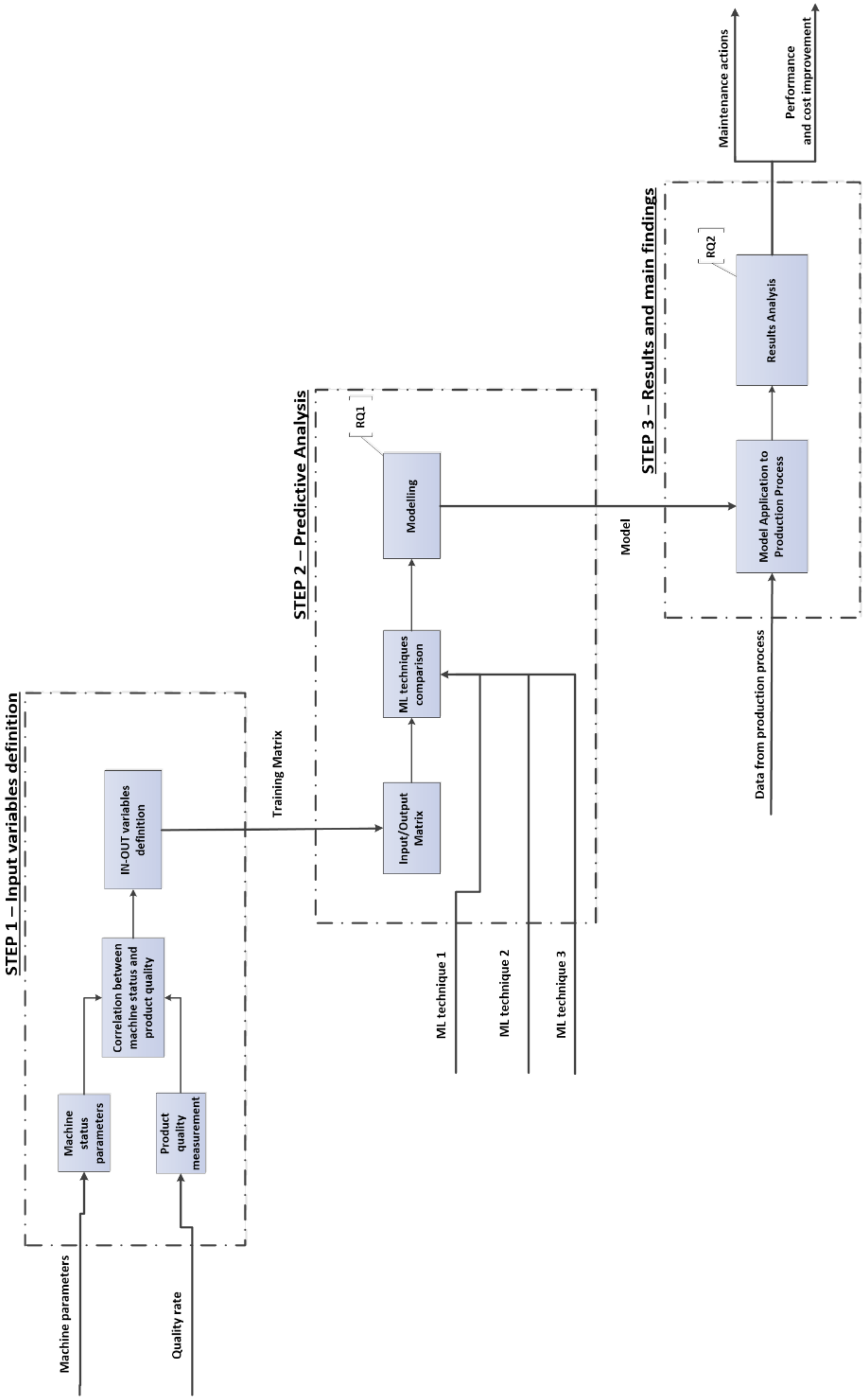


Figure 23: Research Framework

3.1.1 Input variables definition

The first step (Figure 24) aims to define a set of variables concerning the machine status and the PQ according to the application case. in this step is necessary to take into account the machine parameters that can be related both to the continuity of the process and the compliance of the product with the quality requirements, and to define PQ variables that follow in a timely manner the variations in the quality rate of the process.

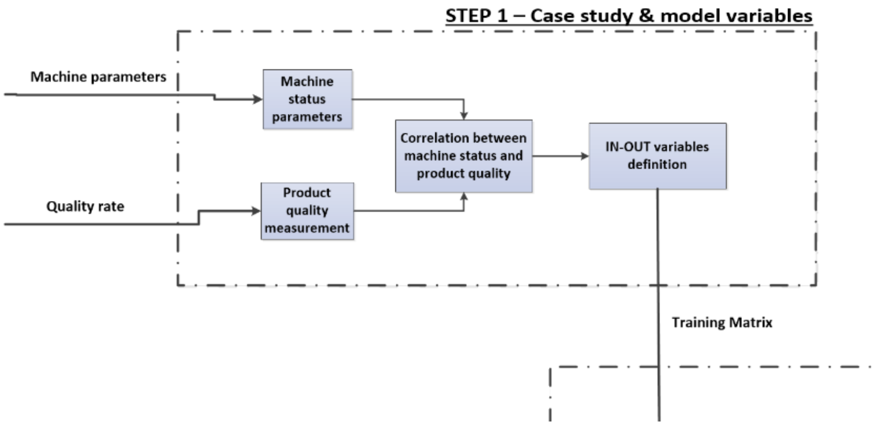


Figure 24: Step 1 in the Research Framework

This step is finalized to relate the PMS with the PQ parameters or, in other words, define which parameters will be the input and output for our model. It is important to define which parameters are important to establish the health of the machine; these can vary depending by machines, industrial sector or products. We can have physical parameters like temperature or pressure, or dynamics parameters like vibrations. For example, the work of Moreno et al. (2009) used torque and speed as input data to monitor the industrial system; instead in the work of Bachraty et al. (2016) the input parameters are depth of cut, cutting speed and feed. The other important variable that we need to define is the quality input variable; this one can be define generally as defected products or can be more detailed by defining products which need to be reworked or products that are wasted. Once all data are analyses, significant Input-Output variables are determined.

3.1.2 The Predictive Analysis Model Definition

Once Input-Output data are defined, in the second step the model structure is defined through combining the monitoring of the variables that represent the machine status, and the prediction of the quality rate, elaborated by artificial intelligence techniques; the model will allow us to select the maintenance strategy according to the desired level of product quality (Figure 25).

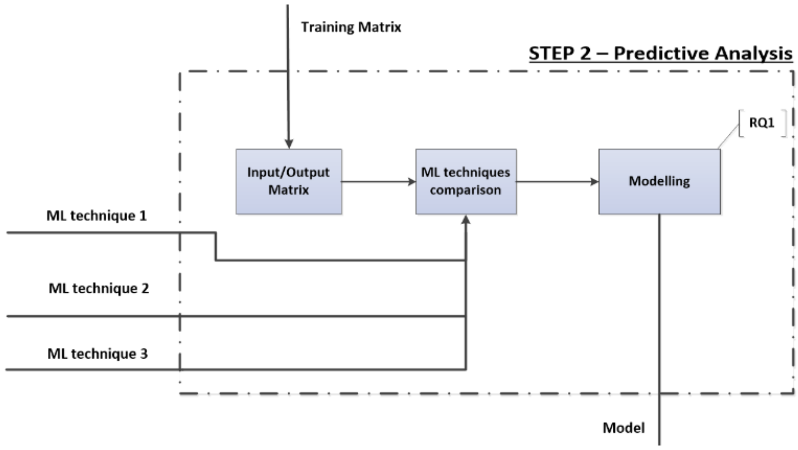


Figure 25: Step 2 in the Research Framework

Different ML techniques are explored to build a model able to learn the behaviour of the system and to detect the potential relations between the performances of the machine and the PQ parameters. In this phase, it is used the comparison of the forecast phase, in terms of accuracy, between different ML techniques, each trained with the same input / output matrix defined in step 1, to validate the algorithm choice. There are many ML algorithms that can be used for these kinds of systems. For example, in the work of Caldas (2015) the Random Forest algorithm is applied to perform predictive maintenance of an electrical substation.; Roque et al. (2022) uses the Gradient Boosting technique to perform failure detection in rotating machines, through machine learning.

Several authors in the last years have proposed maintenance models using Machine Learning. Rivas et al. (2019) elaborate a model to establish the Remaining Useful Life of a machine through a recurrent neural network; Jimenez et al. (2018) present an approach to optimize the sensors in a condition monitoring system employing ultrasonic waves and to classify some features through Machine Learning and Neural Network. Sangje Cho et al. (2018) provides a hybrid machine learning approach combining unsupervised learning and semi-supervised learning to manage the lack of annotations describing the machine status or maintenance history in the data amount available in modern manufacturing companies. Salmaso et al. (2019) use a DOE step before the usual Big Data Analytics and machine learning modeling phase to reduce the difficulty of finding causal relationships among variables.

Anyway, the most used ones are NBC, NNC and ANN.

The NBC represents a supervised learning method as well as a statistical method for classification (Vijaykumar et al., 2014). Assumes an underlying probabilistic model and it allows us to capture uncertainty about the model in a principled way by determining probabilities of the outcomes. This Classification is named after Thomas Bayes (1702-1761), who proposed the Bayes Theorem. Naive-Bayes Classifier is very popular in commercial and open-source anti-spam e-mail filters. (Metsis et. al, 2006).

NNC is a widely used technique for pattern recognition and machine learning (Gagnè and Parizeau, 2007). It is known as a simple yet efficient technique for supervised learning problem with continuous features (attributes); this type of classifier is also often used as a standard system relative to which new classifier systems are compared. The design of NNC requires the specification of three distinct elements: a set of prototype vectors, a classification rule, and a neighborhood proximity measure. The prototypes are representative data used by the classifier to attribute class labels. The NN classification rule consists essentially in choosing the label associated with the nearest neighbor of an unknown input vector.

ANNs consists of processing units in each layer called nodes, connected to the nodes of the next layer and known as fully connected layers. They are also composed of an input layer, an output layer, and one or more hidden layers, and are commonly used for prediction and prognosis using supervised learning (Namuduri et al., 2020).

ANNs have been applied for many years to predictive maintenance systems in various industrial areas. Yam et al (2001) developed a PdM framework to predict flank wear using a posterior propagation neural network. A predictive model for predicting the failure of a heating, ventilation and air conditioning system using artificial neural networks and linear regression was created by Voison et al., (2010). Spendla et al., (2017) in their work adopt artificial neural networks as a predictive model to define the rules necessary to implement predictive maintenance. Recently Lee et al., (2019) to test and monitor machine tool systems, used two powerful classification techniques, support vector machine and artificial neural networks (recurrent neural network and convolutional neural network).

The final output of this step is the model that will be used to monitoring the system and prevent failures.

3.1.3 Validation and results

The final step of Research Framework is the application of the model to the reality and the validation of results (Figure 26).

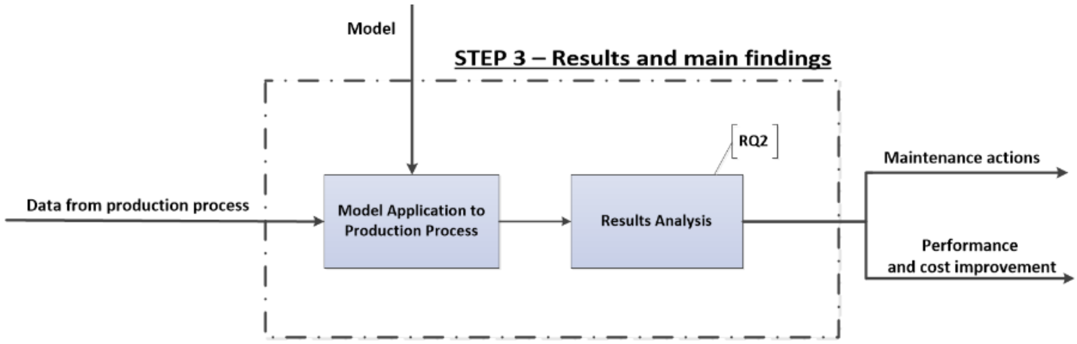


Figure 26: Step 3 in the Research Framework

The input data are based on the values coming from the process; the output will include the maintenance actions suggested by the implemented model and the impacts on the performances deriving from the process in terms of cost, maintenance parameters and quality rate.

4 CASE STUDY AND APPLICATIONS

In this section the research framework is applied to a real case study.

The company LMP srl produces electromechanical breakers, has 10 production lines distributed on 3 different plants in Italy (Figure 27) and 2 sister companies in Bulgarya and United States, and a production capacity of about 1,3 million pcs per year.

Figure 27: An overview of the plant in Naples

The main product categories (Figure 28) are:

- Trip coil
- Magnetothermic actuators
- Electronic components
- Electromagnetic actuators

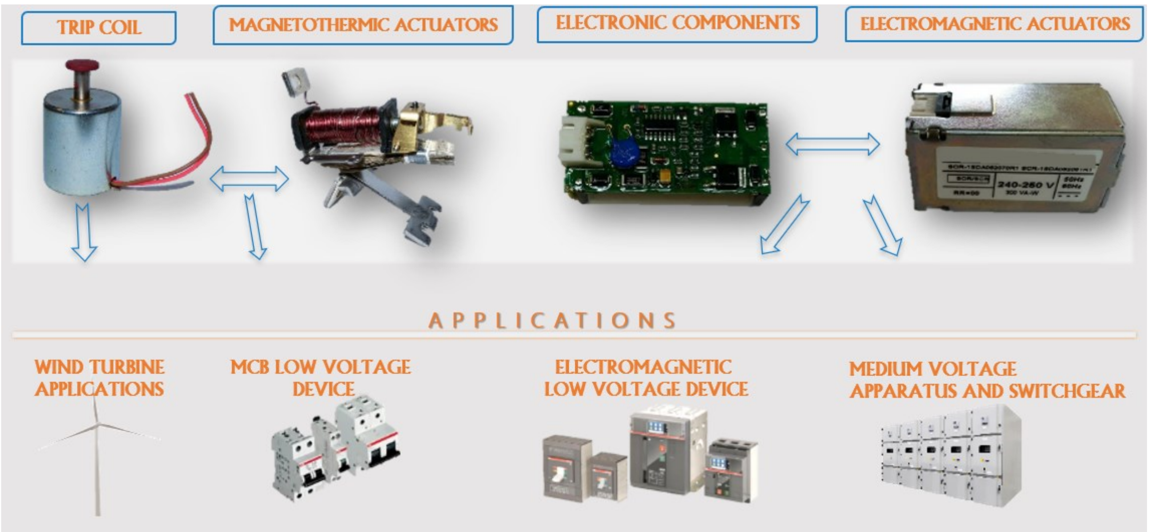


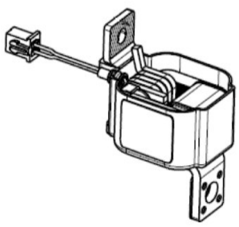
Figure 28: An overview of the products

For the application, a production line for power transformers assembly has been chosen (Figure 29).

Figure 29: The assembly line of our case study

The function of the transformer is to guarantee a constant DC voltage output as an input AC current varies within a certain range, according to the specifications in Table 3:

Table 3: Product specifications

	**Product Specification**	
	In-Out	4 ÷ 7 A → 14.5 ÷ 16.5 V
	Dielectric strength	3.5 KV 50 Hz 2 sec.
	Operating temperature	155/-40 °C (F class according to UL 1446 - IEC 85)

4.1 Input Variables Definition (Step 1)

In this part input variables need to be defined.

The production cycle of the power transformers assembly line has the following operations (Figure 30):

Figure 30: Production flow

1. Assembling of the primary and secondary core in the case
2. Soldering the input cable
3. Application of resin in a vacuum chamber mixer
4. Resin drying
5. Assembling of output connection
6. Insulation test
7. Functional test (check the output current)

In this production process, the vacuum mixer machine is the most critical in terms of product quality rate. This operation concerns the manufacture of a resin layer between the cores of the current transformer that should guarantee the electrical insulation of the product, and it is made by a vacuum mixer machine, which working cycle is made of three phases: the loading, the degassing and the dosing one.

Within this operation the vacuum pressure in the tanks and in the chamber is significant for the success of the phase, and then for the product quality, because the vacuum casting needs to realize an insulating layer without voids, bubbles and/or porosity (Figure 31).

Figure 31: Bubbles on the insulating layer

After the final assembly, the insulation of the product is tested at high voltage according to customer's specifications.

Since the product quality of our case study is related to operation #3, this study is focused on the maintenance in the mixer machine that runs this operation, and specifically of the Vacuum Pumps (VP) that performs the working cycle.

Vacuum pumps are used to reduce the gas pressure in a certain volume and thus the gas density. Consequently, they must remove some gas particles from the volume.

VPs in our mixer machine consist of two primary parts: an electric motor and a vacuum pump. The main factors that can affect the operation of the pump are:

- Oil change: contaminated rotary pump oil is by far the main performance and failure problem. If the oil is discolored or contaminated, the pump may not pull a good vacuum. Failure to change the oil per an appropriate schedule will result in eventual failure that can have several forms, including severe damage to the pump.
- Temperature: too much heat will cause premature pump failure. Excessive heat makes the oil less viscous, so that the pump may not pull a good vacuum. Excessive heat also makes the rubber parts in pump become weak and fail. An excessive external oil leak is often a sign of rubber parts failure.
- Overheated oil: unintended application conditions can make a pump work beyond its capacity and overheat the oil. When the oil is overheated, it often has a burned odor. Overheated oil can harden when it cools, which will lock-up the pump. A frequent cause of overheated oil is a gross vacuum leak. Most of these pumps are not designed to run continuously at higher pressures - typically 10 Torr or greater. The applications that these pumps serve run at lower pressures. A gross vacuum leak can bring the pressure up to a level that is intolerable in the long term. Gross vacuum leaks are common at demountable equipment joints.
- Oil leaks: excessive oil leaks often happen with age. This may mean that the pump needs an overhaul. Running too low on oil will cause disastrous pump failure.
- Intake control: liquids and/or solids entering the pump, especially water, will cause premature pump failure; the water will destroy the pump's internal metal parts due to oxidation. Unavoidable vapor contamination requires more frequent oil changes.

Several authors have addressed the issue of condition monitoring and predictive maintenance of vacuum pumps. Mooney and Shelley (2005) present a summary of new capabilities in pump predictive maintenance using networked monitoring systems. The issue of process by-products amassing in the pumping mechanism was considered by Konishi and Yamasawa (1999). The accumulation of sediments within the operating clearances of the pump causes friction resulting in the pump current surpassing current limits and causing the pump to shut down; an ARMAX model is used to predict vacuum pump motor current was considered. Twiddle et al. (2008) experiment the use of fuzzy-logic based condition monitoring, through with a fuzzy-model based diagnostic scheme to identify mechanical inefficiency and exhaust system obstruction in a dry vacuum pump. It is proven that the power ratios of certain frequency components in the exhaust pressure signal spectrum can be used to predict the gas load, motor current, and hence, mechanical efficiency. Butler et al. (2009) uses artificial neural networks to model the current level of pump degradation using pump process data as inputs, and a double- exponential smoothing prediction method is employed to estimate the RUL (Remaining-Useful-Life) of the pump; the development of a solution benefits from integrating process data, from the upstream processing chamber. A method for vacuum leak detection trough thermography is described by Muhaimin and Ghazali (2015), based on IR thermography image analysis to detect leaks represented by cold spot. Vinogradov and Kostrin (2019) investigate the aspects linked to the oil characteristics to be monitored to establish the correct frequency of oil replacement; a visual colours scale is used to determine the oil condition and the maintenance action. In contrast of this results, our paper tries to use Predictive Maintenance as a methodology to investigate possible pump failures, after the IA model has predicted a deviation in product quality connected to a difference of one or more parameters with respect to the normal operating ranges of the machine.

The machine that performs operation#3 in our production cycle, is equipped with 4 pumps according to the scheme in Figure 32 and the specifications in Table 4:

1. Resin drum pump VP_1
2. Resin tank pump VP_2
3. Hardener tank pump VP_3
4. Dosing chamber pump VP_4

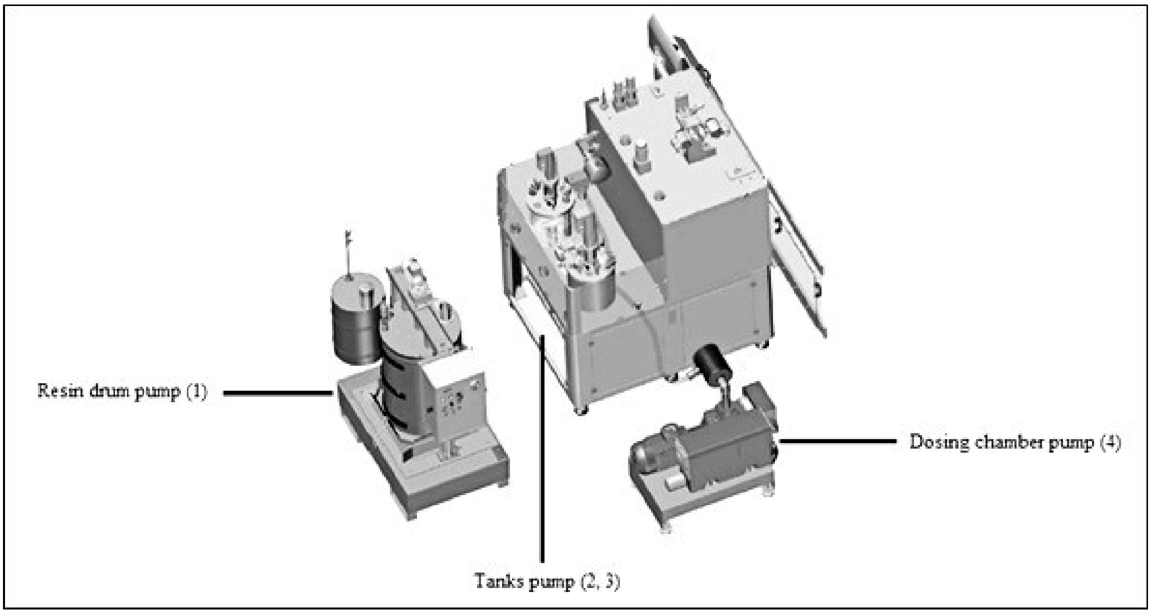


Figure 32: Vacuum pumps in the mixer machine

Table 4: Vacuum pumps specifications

Position	Model	Specification	Picture
1, 2, 3	Becker U 4.20	Oil flooded vacuum pump Flow rate 21 m^3/h, max vacuum 2 mB	
4	Sogevac SV300B	Oil flooded vacuum pump Flow rate 280 m^3/h, max vacuum 800 mB	

The vacuum level in the loading, degassing and dosing phases ensures that the mixture is able to guarantee the insulation performance of the product. The lack of reliability of the pumps could be the cause of insufficient pressure in the tanks and in the dosing chamber, thus affecting the quality of the final product.

The pressures can be monitored by the control panel and are stored in the machine control unit. A typical maintenance indicator for vacuum pumps is the engine temperature that is stored in the machine control unit. Therefore, the parameters that allow to define the machine status are pressure and temperature for each VP, then the inputs of our model are in the Table 5:

Table 5: Machine status input variables

Variable	Description	Unit of measure
p_1	Resin tank pressure	mB
p_2	Hardener tank pressure	mB
p_3	Resin degassing pressure	mB
p_4	Vacuum chamber pressure	mB
T_1	Pump 1 temperature	°C
T_2	Pump 2 temperature	°C
T_3	Pump 3 temperature	°C
T_4	Pump 4 temperature	°C

As PQ variable, the First Pass Yield (FPY) has been chosen; this index represents the percentage of pieces that do not need to be reworked, with respect to the total daily production.

4.2 THE PREDICTIVE ANALYSIS MODEL DEFINITION (STEP 2)

Step 2 concerns the prediction analysis, in which the FPY is estimated on the basis of the input variables, and a decision stage and a decision phase, which suggests a maintenance action based on the output of the previous stage (Figure 33).

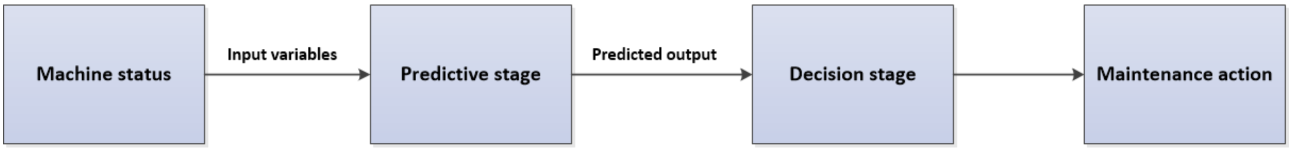


Figure 33: Model structure

To choose the best ML technique to implement the predictive phase, the accuracy of a Naive-Bayes Classifier (NBC), a Nearest Neighbor Classifier (NNC) and an Artificial Neural Network (ANN) was compared. because from the literature review previously analyzed, it emerged that these are the most commonly used techniques for classification problems.

NBC, NNC and ANN have been trained through the Classification Learner App of Matlab with an 8x150 matrix input matrix and a 1x150 output matrix, representing the observation of 150 working days:

$$IN = \begin{bmatrix} p_1^1 & \dots & p_1^{150} \\ \dots & \dots & \dots \\ t_4^1 & \dots & t_4^{150} \end{bmatrix}$$

$$OUT = [FPY^1 \quad \dots \quad FPY^{150}]$$

The pressure values of the pumps and the dosing chamber are stored on the machine control unit, the temperature values of the pumps are read by sensors and stored on the machine control unit, the FPY index is calculated automatically by a line performance software, in which operators enter data on daily production and non-compliant pieces.

The output variable FPY has been classified according to 3 categories:

1. FPY < 90%
2. 90% ≤ FPY ≤ 95%
3. FPY > 95%

Figure 34 shows the Confusion Matrix for the NBC trained through the IN and OUT matrices:

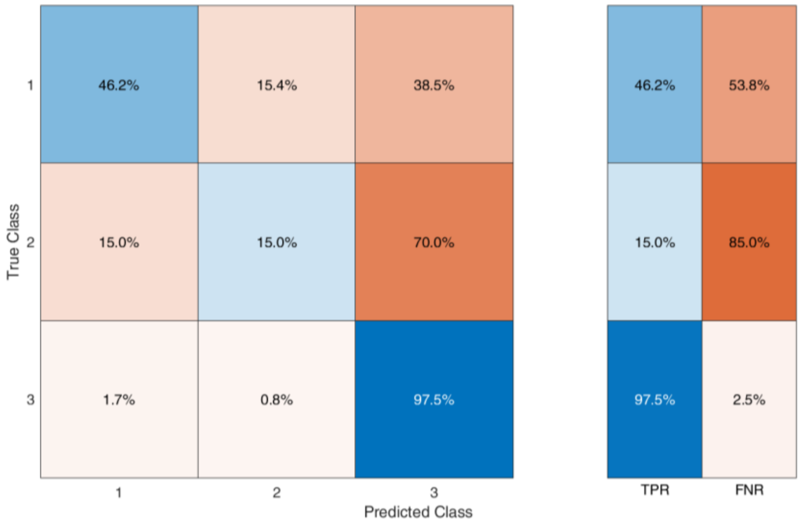


Figure 34: Confusion Matrix for the Naive-Bayes Classifier

In this case, the FNR is high for classes 1 and 2 (representing respectively FPY=2 and FPY=1), this indicates the Naive-Bayes Classifier should not predict the correct output when the input variables should suggest a FPY index in these two classes.

This pattern is visible in the scatter plot (p_4 vs t_4) of Figure 35:

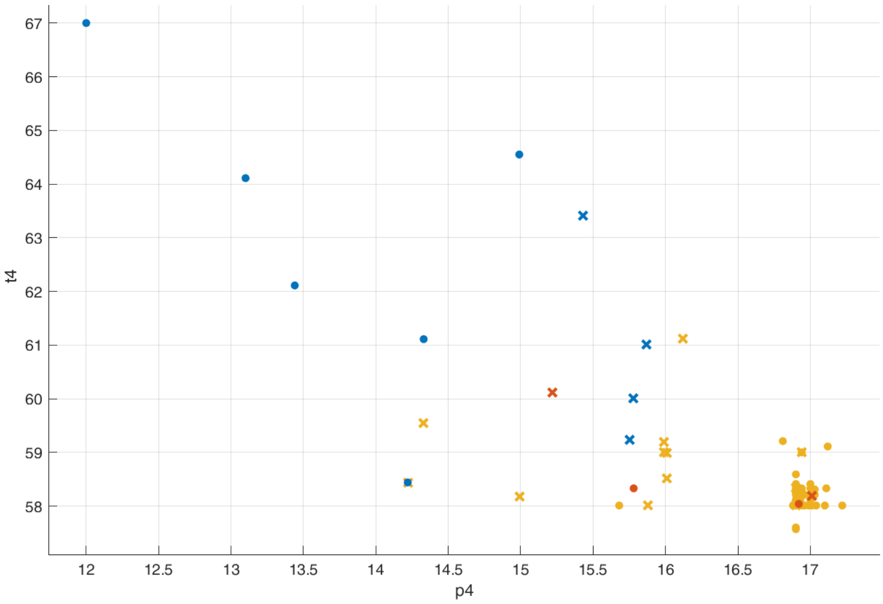


Figure 35: Scatter plot for the Naive-Bayes Classifier (p4 vs t4)

In this plot orange and yellow dots (representing respectively FPY=2 and FPY=1) are overlapped in some areas of the plot (the X dots are incorrect predictions), and this indicates the NNC is unable to predict the right output. The overall accuracy for NBC is 75%.

Figure 36 shows the Confusion Matrix for a NNC (K=1) trained with IN and OUT matrices:

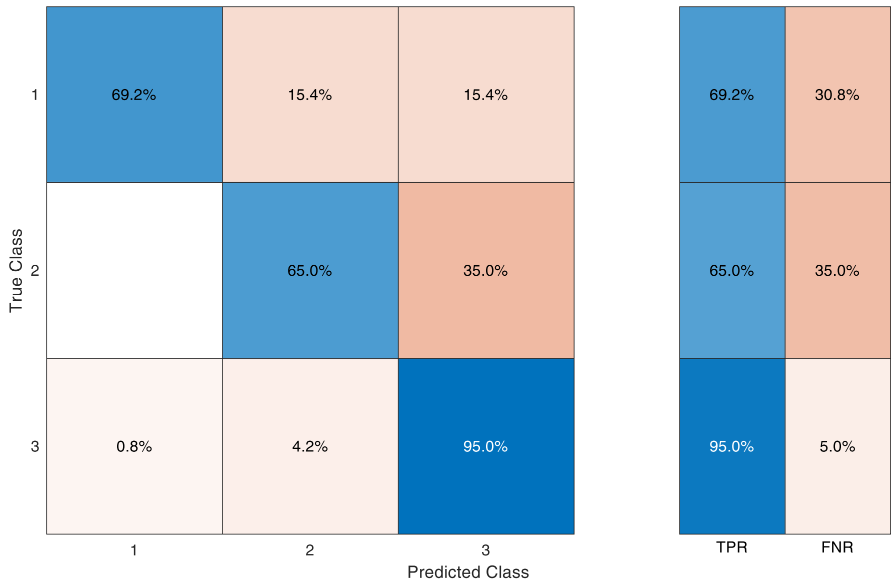


Figure 36: Confusion Matrix for the Nearest Neighbor Classifier

In this case, the FNR is better for classes 1 and 2 (representing respectively FPY=2 and FPY=1), but the accuracy is less in prediction of class 3; the scatter plot (Figure 37) shows overlapping for classes 1 and 2.

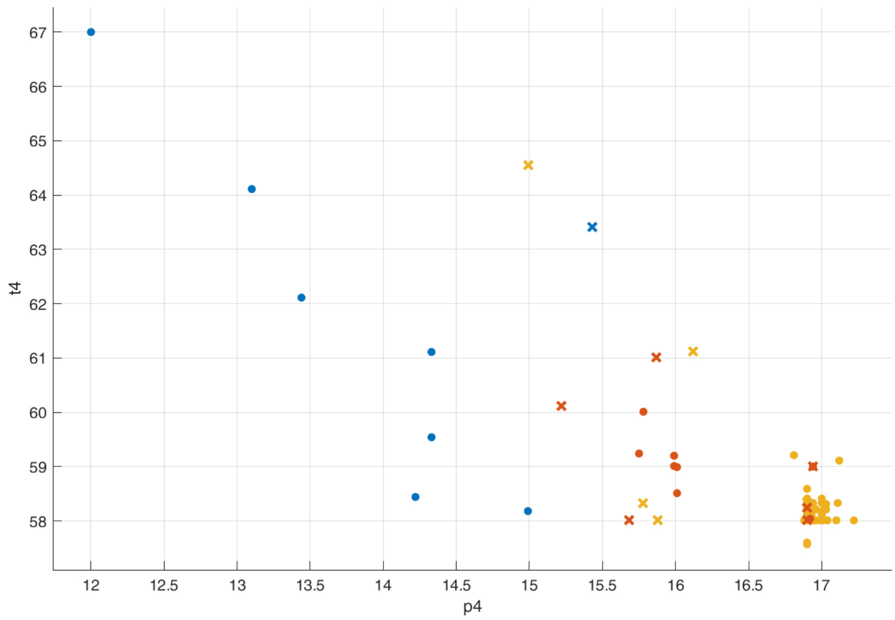


Figure 37: Scatter plot for the Nearest Neighbor Classifier (p4 vs t4)

The overall accuracy for NNC is 88%.

For ANN, the best accuracy result (93%) has been obtained with 17 hidden neurons; the data used to design the NN is divided in following groups:

- 75% for Training set,
- 15% for Validation set
- 15% for Testing set

The regression plots (Figure 38) display the network outputs with respect to targets for training, validation, and test sets. For a complete fit, the data should fall along a 45° line, where the network outputs are equal to the targets.

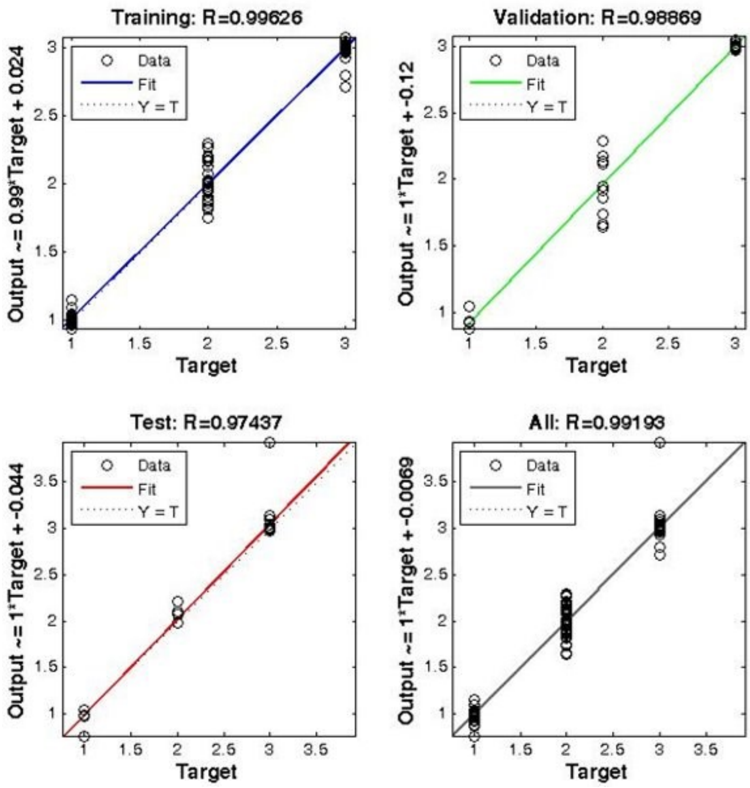


Figure 38: Regression plots for Artificial Neural Network

By comparing the overall accuracy of the Naive-Bayes Classifier and Nearest Neighbor Classifier with that of the ANN (Table 6):

Table 6: Accuracy comparison among AI techniques

Model	**Accuracy**
Naive-Bayes Classifier (NBC)	75%
Nearest Neighbor Classifier (NNC)	88%
Artificial Neural Network (ANN)	93%

the latter was chosen as a predictor of our model.

The predictive phase is completed by combining the artificial neural network (ANN) with a Fuzzy Inference Engine (FIE). The variables describing the machine status are used as input of ANN and FIE, and the predicted FPY (ANN output) represents an additional input to the FIE to determine the appropriate maintenance action for each pump (Figure 39):

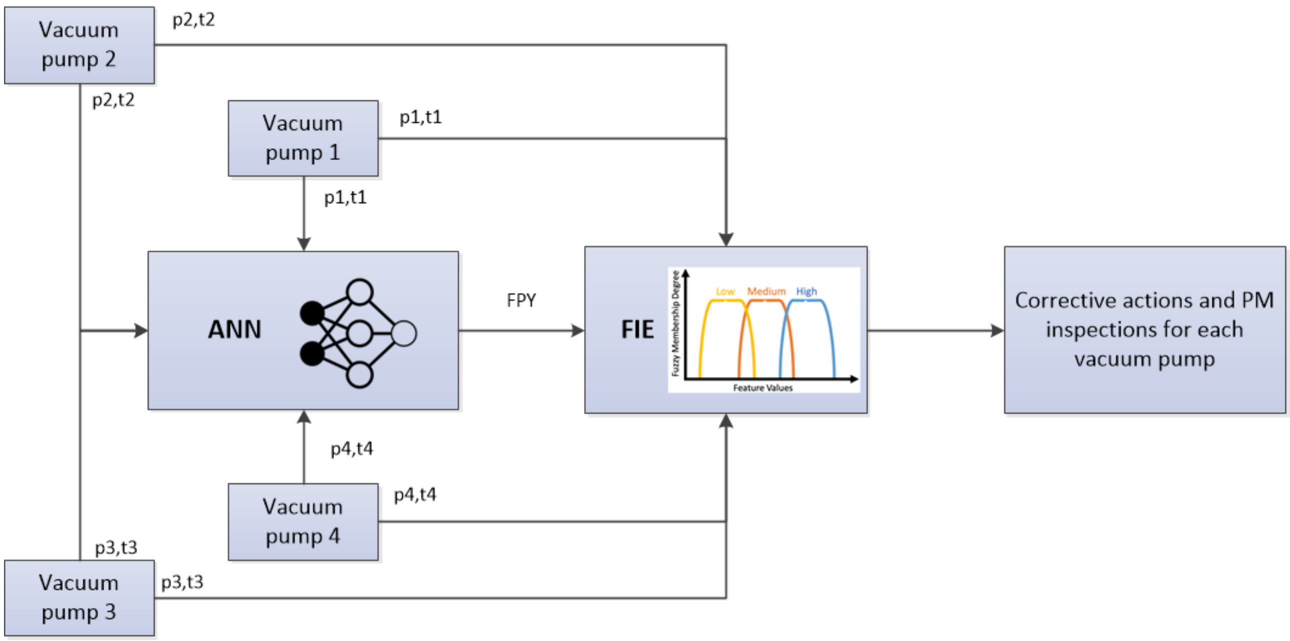


Figure 39: Model flow

In summary, ANN elaborates a prediction, based on the machine status, of the quality rate of the process, and the FIE suggests which corrective action should be carried out on the process according to the ANN output and to the machine status parameters; this strategy uses the ability of the neural networks to predict the behavior of the system, and of the objectivity of the fuzzy rules to decide the action to be taken (Behfarnia and Khademi, 2017).

In details, the FIE receives as input normalized pressure and temperature of each VP, and normalized FPY level estimated by ANN, and elaborates a criticality index IC according to a set of rules (Figure 40):

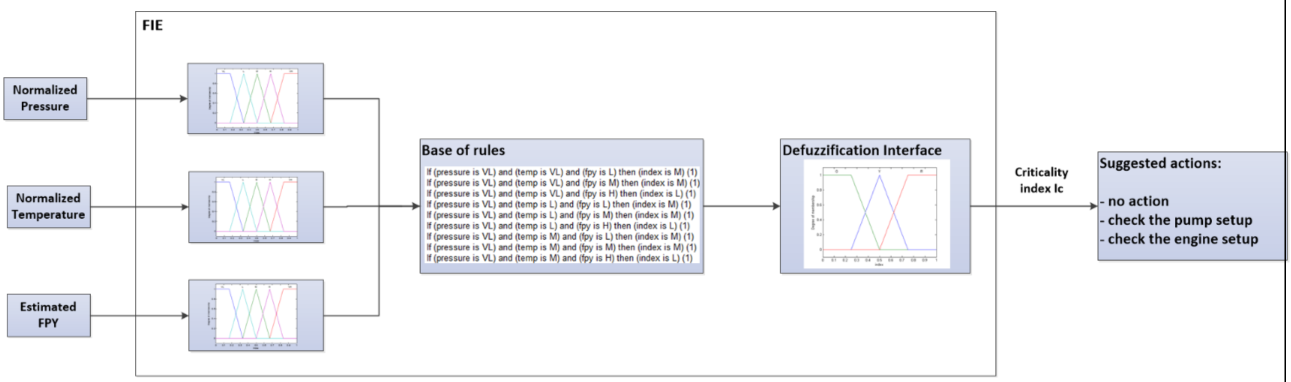


Figure 40: FIE flow

The rules are designed to assess whether the FPY index deviation is caused by human error or an actual anomaly on the pump; in the first case, a low-pressure value (VL), not corresponding to an increase in temperature (L), and an FPY=2 (M), indicates a setup error (I_C =M), as expressed by the following rule:

if (Pressure is VL) and (Temperature is L) and (FPY is M) then (I_C is M)

In the second case, the pressure drop combined with a high temperature (VH), and an FPY= 1, leads to a high I_C (H):

if (Pressure is VL) and (Temperature is VH) and (FPY is L) then (I_C is H)

Through the index value, a corrective action (after defuzzification) is suggested to the user as follows:

- I_C = L → no action
- I_C = M → setup error → check the pump setup
- I_C = H → possible engine failure → check the temperature on vacuum pump engine

This portion of the study responds to the need, expressed through RQ_1, to include the quality parameters of the product, within the problem of predictive maintenance; in fact, the prediction of the FPY is used, together with the operating parameters of the machine, as one of the variables that guides the decision of the model with respect to the choice of maintenance actions.

4.3 RESULTS ANALYSIS (STEP 3)

In Step 3 we analyze how the model predicts the quality rate and suggests a maintenance action through 3 different cases:

1. The parameters of all VP are in the normal operating range:

$$IN = \begin{bmatrix} 8\ mB \\ 7\ mb \\ 8\ mB \\ 18\ mB \\ 28^\circ\ C \\ 29^\circ\ C \\ 28^\circ\ C \\ 45^\circ\ C \end{bmatrix} \dashrightarrow FPY_{estimated} = 3 \dashrightarrow FIE_{indexes} = \begin{bmatrix} 0.33 \\ 0.33 \\ 0.33 \\ 0.39 \end{bmatrix}$$

In this case no action is required on VP_1, VP_2, VP_3, VP_4

2. The pressure of one VP_4 is low:

$$IN = \begin{bmatrix} 8\ mB \\ 7\ mb \\ 8\ mB \\ 10\ mB \\ 28^\circ\ C \\ 29^\circ\ C \\ 28^\circ\ C \\ 45^\circ\ C \end{bmatrix} \dashrightarrow FPY_{estimated} = 2 \dashrightarrow FIE_{indexes} = \begin{bmatrix} 0.33 \\ 0.33 \\ 0.33 \\ 0.67 \end{bmatrix}$$

In this case FIE suggests checking the pressure setup of VP_4 (the low pressure may be caused by operator error)

3. The VP_4 is low, and the temperature is high:

$$IN = \begin{bmatrix} 8\ mB \\ 7\ mb \\ 8\ mB \\ 10\ mB \\ 28^\circ\ C \\ 29^\circ\ C \\ 28^\circ\ C \\ 61^\circ\ C \end{bmatrix} \dashrightarrow FPY_{estimated} = 1 \dashrightarrow FIE_{indexes} = \begin{bmatrix} 0.33 \\ 0.33 \\ 0.33 \\ 0.83 \end{bmatrix}$$

The FIE considers the association between the pressure drop and the temperature increase as an anomaly that heralds a failure on the pump; the suggested action is to inspect the overheated area through a thermographic camera (Figure 41).

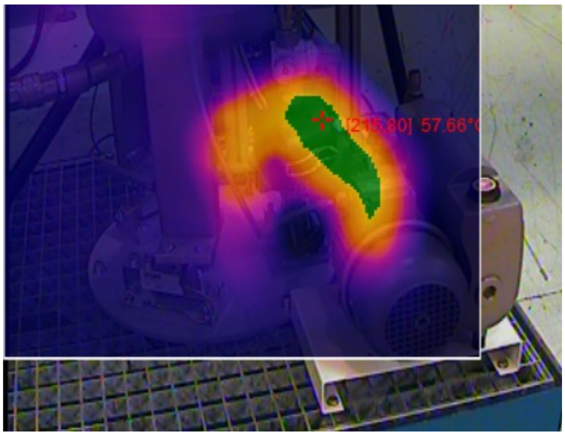


Figure 41: Thermographic inspection of a VP

In case 1, the model evaluates the input data as an optimal situation, in case 2 the low pressure, not being correlated to an increase in engine temperature, is judged as a setup error by the operator. In case 3 the temperature increase is considered as a signal of a possible failure of the pump.

Then, the model was applied to the production process for a period of 6 months, corresponding to an observation period of 150 working days, each with two shifts.
During this period, 46 anomalies have been detected on the 4 vacuum pumps, of which 32 considered to be resolved with a check on the machine parameters and 14 evaluated as possible pump failures.
Table 7 shows the difference between the real events and the predicted events in the 2 categories of anomaly:

Table 7: Real events vs predicted events

	Setup errors	Possible Failures
Predicted	32	14
Real	31	13
Not predicted	0	1

Summarizing, the 96,9% of setup errors was correctly predicted by the model, and 92,8% of possible failures was reported and confirmed by a thermographic inspection on the vacuum pump. In one case the failure even has not been predicted (with serious damage to the pump), because the anomaly had not generated an increase in temperature, and consequently the model was not able to anticipate the failure.
An interesting result concerns the variation of MTBF and FPY by comparing the As-Is process and the period of first application of the model to the production process (Table 8):

Table 8: Process parameters during As-Is phase and model application

	Before PMS	After PMS
Operating time (h)	1248	1313
Number of maintenance events NME	33	27
MTBM	37.8 h	48.6 h
MTTR	168 min	101 min
FPY	96.4 %	98.7%

where MTBM is calculated as (Ben, 2022) in (22):

$$MTBM = \frac{Uptime}{Number\ of\ maintenance\ events} \tag{22}$$

The comparison shows an evident improvement in MTBM (from 37 to 48 h) (Figure 42):

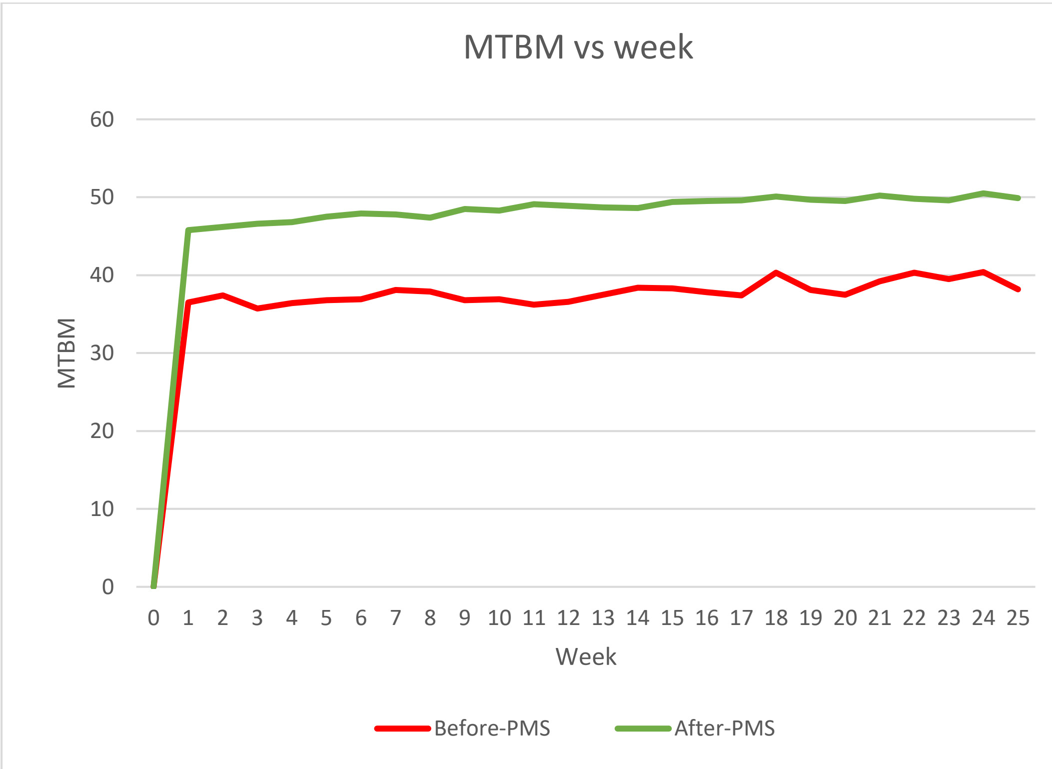


Figure 42: MTBM comparison between data collection and model application

At the same time, an optimization of the FPY index (Figure 43) confirms as the inclusion in the model of a production quality index, which must be monitored and optimized, effectively leads to an improvement in the continuity of operation of the machine, when this index is dependent on some operating parameters of the machine itself (RQ_2).

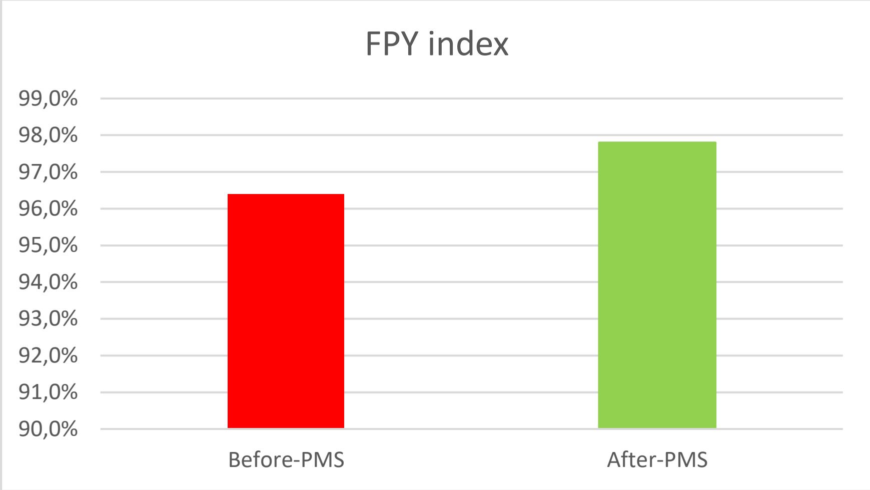


Figure 43: FPY comparison between As-Is phase and model application

4.4 Costs Analysis

Finally, an estimate was made of the total cost $Cost_T$ calculated as the sum of the cost related to the lack of production during machine downtime $Cost_{DT}$, the cost of maintenance intervention $Cost_M$, the cost of the time necessary for the reworking of waste pieces $Cost_S$, and cost to implement the model $Cost_{AI}$, as expressed in (eq. 23):

$$Cost_T = Cost_{DT} + Cost_M + Cost_S + Cost_{AI} \text{ [€/year]} \quad (23)$$

The "Cost of Down time ($Cost_{DT}$)" is expressed as:

$$Cost_{DT} = (K_1 * NME * MTTR) \text{ [€/year]} \quad (24)$$

where K_1 is the missed production hourly cost, NME corresponds to the number of maintenance events in the considered period (usually one year) and the $MTTR$ parameter is the Mean Time To Repair (Table 7).

The maintenance cost $Cost_{MT}$ is calculated as the number of maintenance events happened in the last period multiplied by the average maintenance hourly (K_2) cost and the $MTTR$, and the cost of spare parts $Cost_{SP}$

$$Cost_{MT} = Cost_{MT} + Cost_{SP} = (K_2 * NME * MTTR) + Cost_{SP\_AV} * NME \text{ [€/year]} \quad (25)$$

Where K_2 is the mean cost of maintenance intervention and $Cost_{SP\_AV}$ is the mean cost of the spare sparts.

The scraps cost is evaluated as:

$$Cost_S = (K_3 * (1 - FPY) * YP * RWT) \text{ [€/year]} \quad (26)$$

where K_3 is the manpower hourly cost, YP is the yearly production, FPY is defined as "First Pass Yield" i.e. the percentage of pieces that don't need to be reworked, and finally RWT is the reworking time needed.

Finally, the "Cost of Implementation ($Cost_{AI}$)" is:

$$Cost_{AI} = K_4 + K_5 \qquad [€/year] \qquad (27)$$

where K_4 is the yearly depreciation cost for the implementation of the model (sensors, software, thermographic tools, etc.) and K_5 is the yearly managing cost of the tools.

The value of constants of equations 24-27 are reported in Table 9 and referred to the observation period T= 150 days:

Table 9: Constant values in cost calculation

Constant name	**Value**	
K_1	220 €/h	
K_2	80 €/h	
K_3	29 €/h	
YP	27000 pcs	
RWT	410 sec/piece	
K_4	1.520 €	
K_5	2.160 €	
$Cost_{SP\_AV}$	165 €	
NME	33 (Before-PMS)	27 (After-PMS)
MTBM	37.8 h (Before-PMS)	48.6 h (After-PMS)
MTTR	168 min (Before-PMS)	101 min (After-PMS)
FPY	96.4% (Before-PMS)	98.7% (After-PMS)

In Table 10 the costs of the data collection phase and the period of first application of the model to the production process are reported (*T=150 days*):

Table 10: Cost comparison between As-Is phase and model application

	Before-PMS	After-PMS	Variation %
$Cost_{DT}$ (€/T)	20.291,04	9.980,82	-50,81%
$Cost_{MT}$ (€/T)	7392,00	3639,99	-50,76%
$Cost_{SP}$ (€/T)	5445,00	4455,00	-18,18%
$Cost_{S}$ (€/T)	3.209,55	1.159,00	-63,89%
$Cost_{AI}$ (€/T)	0	3.680,00	/
$Cost_{T}$ (€/T)	36.337,59	22.914,81	-36,93%

The application of the model to the production process allows to save about 50% of cost for machine stops, and 64% of costs for scraps (Figure 44).

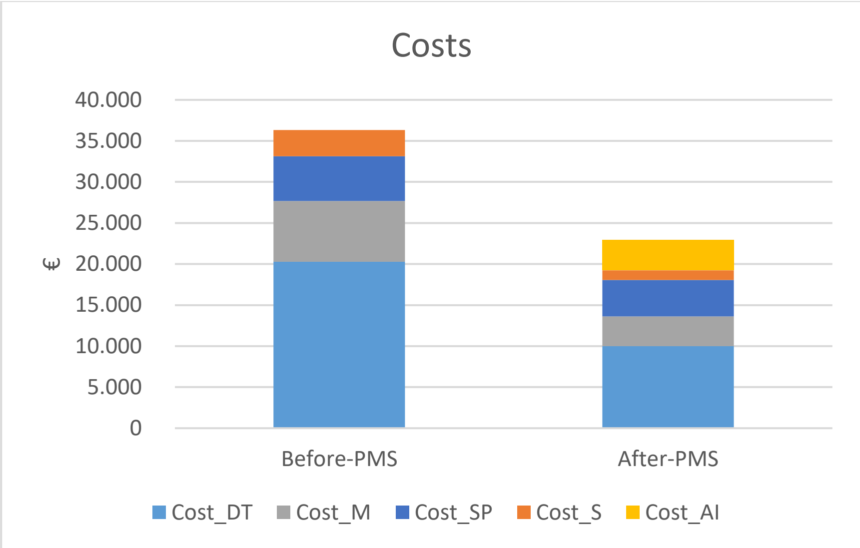


Figure 44: Cost comparison between As-Is phase and model application

A sensitivity analysis has been conducted to find the impact of some parameters on cost function. It is achieved by changing key assumptions made in the evaluation (separately or severally) and recording the impact on the result (output) of the calculation; it provides valuable information to decision-makers about the robustness of their decision based on the findings of an economic evaluation (Orko, 2022).

The total cost was calculated by first changing the annual production YP in a range of ±20%, keeping all the other parameters fixed, and then the number of maintenance events in a range of ±20%; the results are shown in Figure 45.

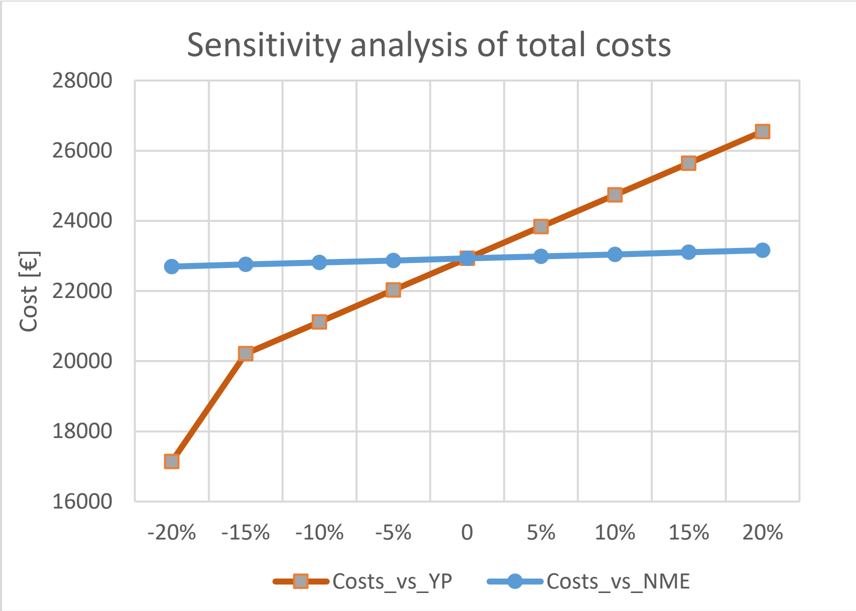


Figure 45: Sensitivity analysis of total costs

The impact of the variation of the annual production capacity on the cost appears to be extremely limited, as the production volume is linked in the cost function only to the scrap cost category $Cost_{SP}$ through the FPY. From a practical point of view, it means that the company will not be affected by significant variations in cost (± 1%) if there are substantial variations in demand, and therefore in production capacity, also in consideration that the starting FPY is high, and that the sensitivity to YP can be even lower if the quality rate is improved further.

Of an opposite nature are the considerations concerning the sensitivity of the cost function to NME; the variation in the number of maintenance events leads to a cost variation range between -25% and +15%. This is due to the direct link that NME has with both $Cost_{DT}$ and with $Cost_{MT}$ and $Cost_{SP}$; therefore the performance of the model in terms of savings for the company strongly depends on how much it is able to limit the number of machine stops, through the setting of a correct predictive maintenance strategy.

4.5 Model Validation

To verify that the developed framework is of general application, it was also tested on a different production process; specifically, it was verified how it works on a CNC turning process within the same company (Figure 46).

Figure 46: CNC lathe

According to the results of the study by Mihail (2017), one of the machine parameters that can affect the quality of the final product (in terms of dimensions and surface roughness) is the cutting speed; based on the results of the experiments reported in this research, variations in the cutting speed impact on the heat gradient between tool and raw material, causing dimensional and roughness variations on the product.
The emulsion system of CNC lathes has the function of delivering cutting oil on the tools to avoid overheating and control tools wear; this system, however, is also one of the components most subject to failures on this type of machines (Gherghe et al, 2021).
Based on these considerations, the model has been trained with a matrix, build through 150 working shifts of observation, that has as input variables the cutting speed S_C and T_C (expressed in percentage override [0-150%]) and the temperature in the cutting chamber, and as output the FPY rate, classified according to 3 categories:

1. FPY < 50%
2. 50% ≤ FPY ≤ 90%
3. FPY > 90%

Also in this application, the model predicts the quality rate and suggests a maintenance action through 3 different cases:

1. S_C and T_C are in the normal range:

$$IN = \begin{bmatrix} S_C = 100\% \\ T_C = 31.5°C \end{bmatrix} \rightarrow FPY_{estimated} = 3 \rightarrow FIE_{indexes} = [0.33] \rightarrow no\ action$$

2. S_C and T_C are high

$$IN = \begin{bmatrix} S_C = 120\% \\ T_C = 38.8\ °C \end{bmatrix} \rightarrow FPY_{estimated} = 2 \rightarrow FIE_{indexes} = [0.66] \rightarrow check\ cutting\ speed$$

3. T_C is high:

$$IN = \begin{bmatrix} S_C = 100\% \\ T_C = 41.1\ °C \end{bmatrix} \rightarrow FPY_{estimated} = 1 \rightarrow FIE_{indexes} = [0.83] \rightarrow check\ emulsion\ system$$

This second application shows how the framework works correctly on a different production process, after having trained the system with a matrix in which the input variables are the system parameters that directly impact on the quality of the final product.
The use of the model on the turning process was carried out for a period of 6 weeks on 5 machines, with the results shown in Table 11.

Table 11: Results of model application to turning process

	Machine 1	**Machine 2**	**Machine 3**	**Machine 4**	**Machine 5**
Anomalies in the emulsion system	2	1	12	0	0
Wrong cutting speed setting	4	5	2	2	1
FPY	84%	88%	75%	98%	96%

Preliminary data show a link between the number of anomalies found in the emulsion system and the decrease in FPY, in particular for machine 3 (which has the highest number of working hours); however, the link between machine performance, product geometry and the type of material processed should be investigated with more data, and it should be taken into account that the machines have different characteristics (speed, bar channel, control module). The analysis must also be completed with the evaluation of the costs.

4.6 DISCUSSION

The application to the production line of electromechanical components allows us to understand how the model created is able to respond to the RQs:

- the relation of machines' data with predictive maintenance indicators and PQ is implemented through a prediction stage of the quality rate based on the machine variables, and a "decisional" stage through fuzzy logic;

- the link between the variables that describe the status of the machine, and the qualitative rate of the production process allows to control maintenance actions based on scraps optimization, achieving an improvement in the operation of the machine. Moreover, this results in an estimated saving of about 50% of the costs for machine downtime and 64% of the costs for scraps.

Finally, a second application to a different production process in the same company was explored, to confirm the generality and versatility of the model

5 CONCLUSIONS

Predictive Maintenance is of course strategic tool to optimize production, reducing machine's downtime and scraps. Respect to Preventive Maintenance, it allowed to maximize the RUL of components, through a continuous monitoring of the production system behavior, reducing not only failures, but also unnecessary maintenance activities. The large use of I4T in industries made it possible to improve considerably PMS application, making available a large amount of data that can be use as input for AI. AI, with the application of different ML algorithm, is a novel and successful tool applied in PdM. An important parameter that can be considered for maintenance prediction is product quality. In fact, machine can continue to work without failures, but produce more and more scraps; this could be a signal of an imminent failure.

In order to combine PQ parameter and PdM, this study has proposed a new framework, aiming to guide practitioners and research in this issue. According to the results of the literature review, the novelty of the approach adopted in this framework is the possibility of managing the process performance in an integrated way both in terms of number and duration of stops and quantity of waste pieces; the identification of a link between the state of the machine and the quality of the pieces produced makes it possible, through the use of ML techniques, to obtain a combined improvement in production continuity and the quality rate, with savings on the overall costs of the production process. The framework has been applied to a real case study, and a costs analysis have been carried out.

Results are encouraging, as it is possible to obtain an indication of the predictive maintenance action to anticipate possible failures related to the state of the machine and the production quality rate, and a saving of about the 37% of actual costs.

The application of the model to the production line leads to the assertion that when there is a link between the variables that describe the operating status of the machine, and the qualitative parameters of the production process, it is possible to control maintenance actions on the basis of optimization of waste, obtaining at the same time an improvement in the operation of the machine and cost optimization.

In the Industry 4.0 context, the model can be used as a tool to improve the maturity of company in data collection and data handling. In fact, the model requires to collect data in a structured way, to identify critical components and, through a data analysis, to based decision on the data; so, according to classification given in Sala et al. (2021), this scenario is an intermediate level of maturity in data handling and allow to jump to and advanced level if the continuous improvement loop is implemented.

5.1 Managerial Implications

This research resulted from industrial activities aimed at developing a framework for the management of maintenance activities in relation to quality targets. The model has been developed through ML, to support Maintenance Manager and Production Manager in planning of maintenance actions according to a real time monitoring of machine parameters and a prediction of quality rate, with the goal of guaranteeing continuity of production flow and low percentage of not-compliant pieces.

From a managerial perspective, this study and its activities have addressed the mutual effect of machine performances and quality requirements based on the process data, supporting the company in evaluating the impact of maintenance costs and quality costs within a single framework.

5.2 Main Limitations and Future Research

Although the results obtained, two main technical limitations affect the framework developed.

The first concerns the choice to compare only some of the Machine Learning techniques available in the second step of the research framework; a more in-depth study of this section would involve an inclusion in the comparison with other ML techniques and an analysis of how the configuration parameters of these techniques affect the behavior of the model in terms of prediction accuracy.

This part could include using supervised learning with both classification (as in the implemented model) and regression techniques; in the first case the algorithm predicts categorical response, in the second case the model predicts continuous responses. A list of supervised techniques is shown in Figure 47 and configuration parameters in Table 12.

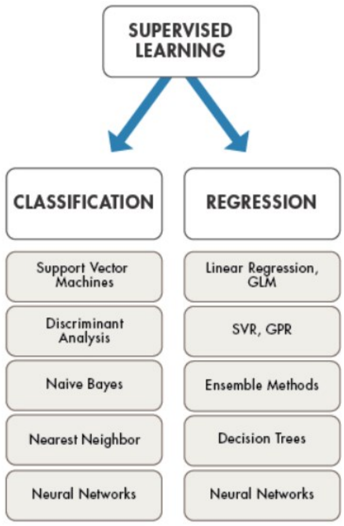


Figure 47: Supervised learning techniques (Matlab Help Center)

Table 12: ML configuration parameters

ML Technique	Main configuration parameters
Decision Tree	Number of splits
Discriminant Analysis	Covariance structure
Naïve Bayes	Distribution
Nearest Neighbor	Number of neighbors
Neural Network	Number of layers
SVM	Type of function (linear, quadratic, gaussian, cubic
Ensemble methods	Ensemble type and number of splits

Similarly, the second limitation is connected to the implementation of the FIE, and in particular to the shape of the fuzzification function; according to Saletic et al. (1999) the response of a fuzzy system and its performance are sensitive to the variations of the shapes of the membership functions, and successive research could evaluate how different choices on the number of labels, on the profile of the input function and on the rules base influence the suggestions coming out of the FIE regarding the maintenance actions.

Moreover, the rule base is the part of the model that needs to be adapted to the type of process to which it is applied; according to Viattchenin et al. (2014), the problem of generation of fuzzy rules is one of more than significant problems in the development of fuzzy classifiers. So, in future research different methods to extract rule can be assessed.

Future research will address the following points:

- the use of additional machine variables that make the prediction of failures more accurate, and can exploit, with a continuous improvement loop, the knowledge generated by the analysis of maintenance interventions data
- the use of ML in Step 1 of Research Framework as tool to detect the correlation between machine parameters and quality rate of the process
- the introduction of a second output variable representing the product performance in the prediction stage
- the application of the framework to different types of production processes.

6 REFERENCES

1. Spendla L., Kebisek M., Tanuska P., Hrcka L. (2017), “Concept of Predictive Maintenance of Production Systems in Accordance with Industry 4.0”, IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI), pp. 405-410
2. Parpala R.C., Iacob R. (2017), “Application of IoT concept on predictive maintenance of industrial equipment”, MATEC Web of Conferences 121, 02008
3. Dreyfus P.A., Kyritsis D. (2018), “A Framework Based on Predictive Maintenance, Zero-Defect Manufacturing and Scheduling Under Uncertainty Tools, to Optimize Production Capacities of High-End Quality Products”, IFIP International Federation for Information Processing, pp. 296-303.
4. Sezer E., David Romero D., Guedeal F., Macchi M., Emmanouilidis C., (2018), “An Industry 4.0-enabled Low-Cost Predictive Maintenance Approach for SMEs: A Use Case Applied to a CNC Turning Centre”, IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC).
5. Changchao Gu, Yihai He, Xiao Han, Zhaoxiang Chen, (2017), “Product quality oriented predictive maintenance strategy for manufacturing systems”, IEEE Prognostics and System Health Management Conference.
6. Carnero M. C., (2006), “An evaluation system of the setting up of predictive maintenance programmes”, Reliability Engineering & System Safety, Volume 91, Issue 8, August 2006, pp. 945-963
7. Kagermann H., Wahlster W., Helbig J., (2013), “Recommendations for implementing the strategic initiative Industrie 4.0”, Frankfurt/Main: Acatech National Academy of Science and Engineering.
8. Leybold - “Fundamentals of Vacuum Technology” - 00.200.02.
9. Mooney M. and Shelley G., (2005), “Data collection and networking capabilities enable pump predictive diagnostic”, Solid State Technology, 48:49–63.
10. Konishi S. and Yamasawa K., (1999), “Diagnostic system to determine the in-service life of dry vacuum pumps”, IEEE Proceedings - Science, Measurement and Technology, 146:270–276.
11. Twiddle A., Jone N.B. and Spurgeon S.K., (2008), “Fuzzy model-based condition monitoring of a dry vacuum pump via time and frequency analysis of the exhaust pressure signal”, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 222:287–293.
12. Butler S.W., Ringwood J.V. and MacGearailt N., (2009), “Prediction of Vacuum Pump Degradation in Semiconductor Processing”, Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes.
13. Muhaimin M.N. and Ghazali K.H., (2015), “IR thermography application for vacuum leak detection of absorption chiller in petrochemical plant”, ARPN Journal of Engineering and Applied Sciences, Vol. 10, No. 22.
14. Vinogradov M. L. and Kostrin D. K., (2019), “Characteristics, Rules of Replacement and Operation of Vacuum Pump Oils” - AIP Conference Proceedings 2089, 020021.
15. Rivas A., Jesus M. Fraile J.M., Chamoso P., Gonzalez-Briones A., Sitton I., and Corchado J.M., (2019), “A Predictive Maintenance Model Using Recurrent Neural Networks”, 14th International Conference on Soft Computing Models in Industrial and Environmental Applications, pp 261-270
16. Jimenez A.A., Gómez Muñoz C.Q., García Márquez F.P., (2018), “Machine Learning and Neural Network for Maintenance Management” International Conference on Management Science and Engineering Management
17. Sangje Cho, May G., Tourkogiorgis I., Perez R., Lazaro O., de la Maza B., Kiritsis D., (2018), “A Hybrid Machine Learning Approach for Predictive Maintenance in Smart Factories of the Future”, IFIP International Federation for Information Processing, pp. 311–317.

18. Salmaso L., Pegoraro L., Giancristofaro R., Ceccato R., Bianchi A., Restello S. and Scarabottolo D., (2019), "Design of experiments and machine learning to improve robustness of predictive maintenance with application to a real case study", Communications in Statistics - Simulation and Computation.
19. Mobley R.K., (2011), "Maintenance Fundamentals, 2. Ed.", Butterworth-Heinemann.
20. Marugán A.P., García Márquez F. P., (2017), "SCADA and Artificial neural networks for maintenance management" - Lecture Notes on Multidisciplinary Industrial Engineering, 2017
21. Ashek-Al-Aziz, Abdullah-Hil Muntakim, (2021), "No relationship between rate of learning and other parameters of Neural Network - a Matlab experience" Journal of Multidisciplinary Engineering Science and Technology
22. Behfarnia K., Khademi F., (2017), "A comprehensive study on the concrete compressive strength estimation using artificial neural network and adaptive neuro-fuzzy inference system", International Journal of Optimization in Civil Engineering
23. Papadopoulos T., Singh S.P., Spanaki K., Gunasekaran A., Dubey R., (2021), "Towards the next Generation of Manufacturing: Implications of Big Data and Digitalization in the Context of Industry 4.0. Prod. Plan. Control
24. Jamwal A., Agrawal R., Sharma M., Kumar A., Kumar V., Garza-Reyes J.A.A., (2021), "Machine Learning Applications for Sustainable Manufacturing: A Bibliometric-Based Review for Future Research" J. Enterp. Inf. Manag. 2021
25. Metsis V., Androutsopoulos I., Paliouras G., (2006), "Spam Filtering with Naive Bayes - Which Naive Bayes?" Conference: CEAS 2006 - The Third Conference on Email and Anti-Spam, July 27-28, 2006, Mountain View, California, USA
26. Vijaykumar V., Tripathy B, T., (2014), "Bayes and Naive Bayes Classifier" Cornell University
27. Gagné C., Parizeau M., (2007) "Coevolution of Nearest Neighbor Classifiers", International Journal of Pattern Recognition and Artificial Intelligence
28. Sala R., Pirola F., Pezzotta G., Vernieri M., (2021), "Improving Maintenance Service Delivery Through Data and Skill-Based Task Allocation", Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems
29. Ben J., (2022), "Implementation of Autonomous Maintenance and its Effect on MTBF, MTTR, and Reliability of a Critical Machine in a Beer Processing Plant", International Journal of Progressive Sciences and Technologies (IJPSAT) - Vol. 31 No. 1 February 2022
30. Pelàez C. E., Bowles J. B., (1994) "Using fuzzy logic for system criticality analysis" Proceedings of Annual Reliability and Maintainability Symposium 449-445.
31. Zhang J., Ran H., Sun L., Guo Y., (2006) "A fuzzy method for risk quantitative analysis of system failure" Proceedings of the Fifth International Conference on Machine and Cyberenetics 1702-1707.
32. Bowles J. B., (2003) "An assessment of RPN prioritization in a Failure Mode Effects and Criticality Analysis" Proceedings of Annual Reliability and Maintainability Symposium 380-386.
33. Bukowski L., Feliks J. (2005) "Application of fuzzy sets in evaluation of failure likelihood" Proceedings of the 18th International Conference on System Engeenering.
34. Papadias B. C., Kaminaris S. D., Machias A. V., (1993) "Substation maintenance using fuzzy sets theory and expert system methodology" Proceedings Joint International Power Conference 2: 611-615.
35. Cooper J. A., (1994) "Fuzzy algebra uncertainty analysis for abnormal-environment safetv assessment" IEEE International Conference on Neural Networks 4: 2166-2171.
36. Klim Z. H., (2004) "Preliminary hazard analysis for the design alternatives based on fuzzy methodology" Fuzzy Information Processing Society 1: 46-50.
37. Hathout I., (2006) "Maintenance prioritization of exisisting transmission lines using priority risk indices (PRI)" 9th International Conference on Probabilistic method Applied to Power Systems.

38. Cassel G., Canha L., Abaide A., Farret F., Daza E., (2007) “Automatic priority of the maintenance activities in distribution systems using multicriterial analysis and fuzzy techniques” POWERENG 2007 237-242.

39. Zadeh L. A., (1988) “Fuzzy Logic” IEEE Computer 21: 83-93.

40. Mamdani E. H., Assilian S. (1974) “A case study of the application of fuzzy set theory to automatic control” Proceedings of IFAC Stochastic Control Symposium.

41. Davidson V., Chu T., Ryks J., (1999) “Fuzzy methods for automated inspection of food products” 18th International Conference of the North American Fuzzy Information Society 909-913.

42. Takagi T., Tajima M., (2001) “Query expansion using conceptual fuzzy sets for search engine” The 10th IEEE International Conference on Fuzzy Systems 3: 1303-1308.

43. Tsuen-Ho Hsu, (2001) “Applying fuzzy set theory to measuring air transportation service quality” 20th NAFIPS International Conference 5: 2864-2869.

44. Liang Cai, Hongkun Chen, (2002) “Fuzzy studies on power quality-part I: index and evaluation” 10th International Conference on Harmonics and Quality of Power 2: 414-418.

45. Awais M. M., Habib-ur-Rehman, (2003) “Recognition of Arabic phonemes using fuzzy rule vase system” 7th International Multi Topic Conference 367-368.

46. Jun-Jian Qiao, Xin-Wu Zhen, Yan-Rui Zhang, (2008) “The application of fuzzy comprehensive evaluation on the water quality on Changjiang river” International Conference on Machine Learning and Cybernetics 3: 1467-1473.

47. Klir G. J., Yuan B., (1995) “Fuzzy sets and fuzzy logic, theory and applications” Prentice Hall.

48. Zadeh L. A. (1996) “Fuzzy Logic = Computing with Words” IEEE Transactions on Fuzzy Systems 4: 103-111.

49. Kayzoglu, T., (2001), “An Investigation of the Design and Use of Feed-Forward Artificial Neural Networks” - Thesis submitted to the University of Nottingham

50. Rumelhart D. E., (1986) “Sequential Thought Processes in PDP Models” Computer Science

51. LeCun Y., Boser B., Denker J.S., Henderson D., Howard R.E., Hubbard W., Jackel L. D., (1989), “Backpropagation Applied to Handwritten Zip Code Recognition”.

52. Hájek, P., (2013), Metamathematics of fuzzy logic (Vol. 4). Springer Science & Business Media.

53. Gherghea I. C., Bungau C., Indre C.I., Negrau D.C., (2021), “Enhancing Productivity of CNC Machines by Total Productive Maintenance (TPM) implementation. A Case Study” The Annual Session Of Scientific Papers - IMT Oradea 2021, IOP Conf. Series: Materials Science and Engineering 1169 (2021) 012035

54. Mihail L. A., (2017) “Robust Development of a CNC Turning Process”, MATEC Web of Conferences 94, (2017)

55. Florian, E., Sgarbossa, F., & Zennaro, I. (2021), “Machine learning-based predictive maintenance: A cost-oriented model for implementation” International Journal of Production Economics, 236, 108114.

56. Krishnamurthy L., Adler R., Buonadonna P., Chhabra J., Flanigan M., Kushalnagar N., Nachman L. and Yarvis, M., (2005), “Design and deployment of industrial sensor networks: Experiences from a semiconductor plant and the North Sea”, in SenSys 2005 - Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems, pp. 64-75.

57. Martin K. F., (1994), “A review by discussion of condition monitoring and fault diagnosis in machine tools” International Journal of Machine Tools and Manufacture, Vol. 34, No. 4, pp. 527–551.

58. Jardine A. K. S., Lin D. and Banjevic D., (2006), “A review on machinery diagnostics and prognostics implementing condition-based maintenance” Mechanical Systems and Signal Processing, Vol. 20, No.7, pp. 1483–1510.

59. Wagner C., Saalmann P., Hellingrat B., (2016), “An Overview of Useful Data and Analyzing Techniques for Improved Multivariate Diagnostics and Prognostics in Condition-Based Maintenance” Proceedings of Annual Conference of the Prognostics and Health Management Society, October 3-6, Denver, Colorado, US.

60. Hofmann T., Schölkopf B. and Smola A. J., (2008), “Kernel methods in machine learning” Annals of Statistics, pp. 1171-1220.

61. Kuo Y., (2006), “Optimal adaptive control policy for joint machine maintenance and product quality control” European Journal of Operational Research, 171(2), 586-597.

62. Ross S. M., (1971), “Quality control under Markovian deterioration” Management Science, 17(9), 587-596.

63. Baumeister R. F., Leary M. R., (1997), “Writing narrative literature reviews” Review of General Psychology, 1, 311–320

64. Webster J., & Watson R. T., (2002), “Analyzing the past to prepare for the future: Writing a literature review” Management Information Systems Quarterly, 26, 3.

65. Snyder H., (2019), “Literature review as a research methodology: An overview and guidelines” Journal of Business Research 104 (2019) 333–339

66. Liberati, A., Altman, D. G., Tetzlaff, J., Mulrow, C., Gøtzsche, P. C., Ioannidis, J. P. A., Moher, D., (2009), “The PRISMA statement for reporting systematic reviews and metaanalyses of studies that evaluate health care interventions: Explanation and elaboration” Annals of Internal Medicine, 151, W–65

67. Moher, D., Liberati, A., Tetzlaff, J., Altman, D. G., (2009), “Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement” Annals of Internal Medicine, 151, 264–269

68. Xiao Y. and Watson M., (2019), “Guidance on Conducting a Systematic Literature Review” Journal of Planning Education and Research 2019, Vol. 39(1) 93–112

69. Kitchenham B. and Stuart C., (2007) “Guidelines for Performing Systematic Literature Reviews in Software Engineering.” In EBSE Technical Report, Software Engineering Group, School of Computer Science and Mathematics, Keele University, Department of Computer Science, University of Durham.

70. Gates S., (2002) “Review of Methodology of Quantitative Reviews Using Meta-analysis in Ecology.” Journal of Animal Ecology 71 (4): 547–57.

71. Petticrew M., and Roberts H., (2006) “Systematic Reviews in the Social Sciences: A Practical Guide. Oxford: Blackwell.

72. Rowley J. and Slack F., (2004) “Conducting a Literature Review” Management Research News 27 (6): 31–39.

73. Lucas P. J., Baird J., Arai L., Law C. and Roberts H.M., (2007) “Worked Examples of Alternative Methods for the Synthesis of Qualitative and Quantitative Research in Systematic Reviews” BMC Medical Research Methodology 7 (1): 4.

74. Okoli C. and Schabram K., (2010) “A Guide to Conducting a Systematic Literature Review of Information Systems Research” Sprouts: Working Papers on Information Systems 10 (26)

75. Noordzij M., Hooft L., Dekker F. W., Zoccali C. and Jager K. J., (2009) “Systematic Reviews and Meta-analyses: When They Are Useful and When to Be Careful.” Kidney International 76 (11): 1130–36.

76. Moreno R., Pintado P., Chicharro J.M., Morales A.L. and Nieto A.J., (2009) “Methodology for evaluating neural networks inputs for gear fault detection” Proceedings of the 2009 IEEE International Conference on Mechatronics, Málaga, Spain, April 2009.

77. Bachratya M., Tolnaya M., Kovacb P., Pucovsky V., (2016) “Neural Network Modeling of Cutting Fluid Impact on Energy Consumption during Turning” Tribology in Industry Vol. 38, No. 2 (2016) 149-155

78. Caldas A.N., (2015) “Desenvolvimento de um Sistema de Apoio `a Decis˜ao para a manuten¸c˜ao preditiva dos ativos de uma subesta¸cao eletrica” Master’s thesis, Faculdade de Eng. da Universidade do Porto, FEUP, Portugal.

79. Roque A.S., Krebs V.W, Figueiro I.C. and Nasser Jazdi N., (2022) “An analysis of machine learning algorithms in rotating machines maintenance” IFAC-PapersOnLine Volume 55, Issue 2, 2022, Pages 252-257

80. Jong-Ho Shin and Hong-Bae Jun, (2015) “On condition-based maintenance policy” Journal of Computational Design and Engineering 2 (2015) 119–127

81. Garg A. and Deshmukh S., (2006) “Maintenance management: Literature review and directions” Journal of Quality in Maintenance Engineering, vol. 12, no. 3, pp. 205–238, 2006, cited By 354.

82. He Y., Gu C., Chen Z., & Han X., (2017) “Integrated predictive maintenance strategy for manufacturing systems by combining quality control and mission reliability analysis” International Journal of Production Research, 55(19), 5841-5862.

83. Rivera-Gómez H., Gharbi A., & Kenné, J. P., (2013) “Joint production and major maintenance planning policy of a manufacturing system with deteriorating quality” International Journal of Production Economics, 146(2), 575-587.

84. Ji-Wen S., Li-feng X., Shi-chang D., & Er-shun P., (2010). “Tool maintenance optimization for multi-station machining systems with economic consideration of quality loss and obsolescence” Robotics and Computer-Integrated Manufacturing, 26(2), 145-155.

85. Tambe P. P., Kulkarni M. S., (2015) “A superimposition-based approach for maintenance and quality plan optimization with production schedule, availability, repair time and detection time constraints for a single machine” Journal of Manufacturing Systems, 37, 17-32.

86. Lu B., Zhou X., & Li Y., (2016) “Joint modeling of preventive maintenance and quality improvement for deteriorating single-machine manufacturing systems” Computers & Industrial Engineering, 91, 188-196.

87. Chen Y., Jin J., (2005) “Quality-reliability chain modeling for system-reliability analysis of complex manufacturing processes” IEEE Transactions on Reliability, 54(3), 475-488.

88. Colledani M., Tolio, T., (2012) “Integrated quality, production logistics and maintenance analysis of multi-stage asynchronous manufacturing systems with degrading machines” CIRP annals, 61(1), 455-458.

89. Saletic D., Velasevic, D.M., Mastorakis, N. E., (1999) “The Influence of Shapes of Fuzzy Sets Membership Functions on Fuzzy System Performances”

90. Viattchenin D.A., Shyrai S., Damaratski A., (2014), “Labeling Consequents of Fuzzy Rules Constructed by Using Heuristic Algorithms of Possibilistic Clustering” Database Systems Journal vol. V, no. 3/2014

91. Menanno M., Savino M.M., Ciarapica F.E., (2021) "Exploring continuous improvement for safety management systems through artificial neural networks", International Journal of Product Development

92. Dolgui A., Bernard A., Lemoin D., Von Cieminski G., Romero D., (2021), "Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems", Springer Science and Business Media LLC

93. Savino M.M., Battini D., Riccio C., (2017) “Visual management and artificial intelligence integrated in a new fuzzy-based full body postural assessment” Computers & Industrial Engineering

94. Orko, S., (2022) “Sensitivity analysis (SA)” Conference: Health finance and cost benefit analysis

7 APPENDIX A – LITERATURE SUMMARY

Domain /Area	Subject	Main Goal	References
Industry 4.0	Maintenance	Concepts of Preventive Maintenance in Industry 4.0	Spendla et al. (2017)
		Application of IoT concept on predictive maintenance	Parpala et. al. (2017)
	The forth industrial revolution	Implementation of strategies for Industry 4.0	Kagermann et al. (2013) Papadopoulos et al. (2021)
Maintenance	Basic concepts	Fundamental of maintenance	Mobley (2011)
	Quality	Maintenance oriented to product quality	Changchao et al. 2017 Dreyfus et al. (2018)
	Vacuum pump	Maintenance and diagnostic	Mooney et al. (2005) Konishi et al. (1999) Butler et al. 2009 Vinogradov et al. (2019)
		Predictive maintenance	Muhaimin et al. (2015)
	Maintenance data	Autonomous maintenance and performances	Sala et al. (2021) Ben (2022)
Machine Learning	Matlab	Matlab and Machine Learning	Aziz et al. (2021)
	Maintenance	Maintenance and neural network	Rivas et al. (2019) Jimenez et al. (2018) Sangje Cho et al (2018) Salmaso et al (2019) Marugán et al (2017)
	Machine Learning	Neural network and adaptive neuro-fuzzy inference system	Behfarnia et al. (2017)
		Naive Bayes classifier	Metsis et al. (2006) Vijaykumar et al. (2014)
		Nearest Neighbor classifiers	Gagné et al. (2007)