IEEE Open Journal of the
Industrial Electronics Society

Received 18 August 2024; revised 2 October 2024; accepted 16 October 2024. Date of publication 21 October 2024;
date of current version 31 October 2024. The review of this article was arranged by Associate Editor Stamatis Karnouskos.

Digital Object Identifier 10.1109/OJIES.2024.3484006

A Detailed Study on Algorithms for Predictive Maintenance in Smart Manufacturing: Chip Form Classification Using Edge Machine Learning

ALESSIA LAZZARO [1], DORIANA MARILENA D'ADDONA [2], AND MASSIMO MERENDA [1,3] (Member, IEEE)

[1]Department of Information Engineering, Infrastructure and Sustainable Energy (DIIES), University Mediterranea of Reggio Calabria, 89124 Reggio Calabria, Italy
[2]Department of Chemical, Materials and Industrial Production, University of Naples Federico II, 80138 Naples, Italy
[3]HWA srl-Spin Off dell'Università Mediterranea di Reggio Calabria, 89126 Reggio Calabria, Italy

CORRESPONDING AUTHOR: MASSIMO MERENDA (e-mail: massimo.merenda@unirc.it).

ABSTRACT Industrial and technological evolution has led to the identification of different techniques and strategies that can best adapt to the needs of Manufacturing Industry 4.0. As industrial production has become more automated, the need for more efficient maintenance strategies has increased. Today, among the possible, several applications demonstrate how the Predictive Maintenance (PdM) strategy is the best performing. In fact, PdM makes it possible to predict an impending failure with high accuracy in order to intervene before failure occurs. This work focuses on the application of PdM technique in order to predict the type of chips produced by a lathe through a machine learning algorithm. Moreover, being our application a delay-sensitive one, to drastically decrease the time delay in prediction, our solution proposes the combination of PdM with the Edge Computing paradigm. To simulate this paradigm, the chosen machine learning models were deployed on STM microcontrollers obtaining both high accuracy (98%) and an inference time in the order of milliseconds.

INDEX TERMS Chip form classification, cyber-physical system (CPS), edge computing (EC), Industry 4.0, industrial systems, manufacturing, predictive maintenance (PdM), supervised learning, turning.

NOMENCLATURE

ADC	Analogue-to-digital converter.
AI	Artificial intelligence.
ANN	Artificial neural network.
BNN	Bayesian neural network.
CFS	Cutting force signal.
CPS	Cyber-physical system.
DIPF	Design-installation-potential failure-failure.
EC	Edge computing.
EdgeAI	Edge intelligence.
FFBP	Feed-forward back propagation.
I2C	Inter-integrated circuit.
I4.0	Fourth industrial revolution.
I5.0	Fifth industrial revolution.
IoT	Internet of Things.
KNN	k-Nearest neighbors.
LPA	Linear predictive analysis.
MACC	Multiply-accumulate per cycle.
ML	Machine learning.
MLOps	Machine learning operations.
MLP	Multilayer perceptron.
MVR	Multiple variable regression.
NN	Neural network.
PCA	Principal component analysis.
PdM	Predictive maintenance.
QoS	Quality of service.
RA	Regression analysis.
ReLU	Rectified linear unit.
RF	Random forest.
RUL	Remaining useful life.
RTF	Run-to-failure.
SOM	Self-organizing map.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see
https://creativecommons.org/licenses/by-nc-nd/4.0/

SPI Serial peripheral interface.
SVM Support vector machine.
SVR Support vector regression.
TBM Time-based maintenance.

I. INTRODUCTION

The development of the fourth industrial revolution (I4.0) has disrupted the entire industrial sector in order to meet new market demands. Since state and market borders are deleted, comprehensive globalization started to rule and the demand and supply of products is greater than ever. In detail, the rapid development of technology and informatics have brought major changes in the market, causing the abandonment of the classical production methods [1].

The solution deployed by I4.0 is called Smart Factory. Based on the third industrial revolution initiative, this solution consists of the integration of all recent IoT technological advances in computer networks, data integration, and analytics to bring transparency to all manufacturing factories [2].

In order to make the industrial conversion, the manufacturing industry has to deal with multifaceted challenges because investments are required not only at the production level, to increase economic and environmental efficiency, but also at the organizational, strategic, business, and customer service levels.

These needs create a space for the search of a new paradigm. In particular, the CPS finds applications in manufacturing as it supports the production process thanks to the interaction between the physical word and the internet (cyber) space, hence the name of the system. The immediate consequence is an increase in the level of complexity of the production process, but provides significant benefits for the entire manufacturing industry.

In general, architecture for CPS in Industry 4.0 manufacturing systems consists of two main functional components: 1) the advanced connectivity that ensures real-time data acquisition from the physical world and information feedback from the cyber space; and 2) intelligent data management, analytics and computational capability that constructs the cyber space [3].

From this point of view, physical processes are the plant that is controlled by a cyber system. Among the numerous advantages, encapsulated in the term "global optimization of systems," one of the most important is enabling monitoring, which increases control over the entire work chain, thereby improving decision-making capacity especially during the occurrence of unforeseen events.

Keeping changing states under control during the production phase is one of the challenges that modern companies face. In fact, to survive in the global competitive marketplace, it is required to improve the quality of the product and, at the same time, to shorten production cycle, reducing equipment downtime, and lowers production costs, thus decreasing maintenance costs.

This is because, besides being the main component in time and cost calculations, maintenance plays an important role in intelligent systems. Indeed, reliability and safety are challenged by a highly complex, automated, and flexible industrial system [4].

Nowadays more companies are producing the Big Data generated by the numerous sensors, for real-time data acquisition. The analysis of industrial Big Data can provide new solutions for maintenance, ensuring the improvement of system reliability by achieving almost zero downtime.

These needs are leading to terrific attention on maintenance strategies and today, among those available, PdM is proving to be the preferable option. Several studies show that the application of PdM in the company has as its main benefit the reduction of production costs (between 15% and 70% [5]) as well as the advantage of having greater safety for employees, the company, and the environment. In fact, thanks to the evolution of technologies such as AI, especially ML methods, and the IoT platform, PdM allows the real-time health monitoring of the assets to predict possible failures and replace the components just before their breakage, increasing the success probability of the mission and decreasing the time-to-market.

Moreover, one of the requirements of a Smart Factory is Real-time capability. This refers to the ability of the system to respond to changes on time, such as changes in the status of the internal production system (e.g., malfunctions and resource failures). This means that responses to internal changes, monitoring, and controlling should be in real time. Disturbances should be detected on time, and the system should have the ability to recover rapidly [6], [7].

In most of the current state of the art, the responsiveness feature is implemented with the Cloud Computing paradigm as a control center for processing data, gaining knowledge, and predicting the state of health of equipment.

An innovative and interesting solution is achieved with the EC paradigm, in which sensors and edge devices distribute the data processing work to each other, thus reducing data transmission costs. Consequently, adding the edge layer increases the relationship between the physical factory floor and the cyber decision space, enabling immediate local decision-making even in those time-sensitive applications where the latency of the Cloud is too high to support decision-making based on PdM analytics.

This study contributes to bridging the gap of AI-based Edge PdM by examining its benefits. In detail, it aims to compare several supervised ML algorithms, placed on a microcontroller, for predicting the health of the working environment. This article extends our previous work [8] with an in-depth case study on three microcontrollers and an analysis of the implications of edge device performance.

The following work presents a method in order to find out the ML model that better fits a practical example. In this sense, it focuses on the application of ML models to carry out the PdM of a turning machine, classifying the shape of the chip on the basis of the forces applied by the tools on the material. Indeed, the formation of too-long chips may complicate the machining process, as a chip that is too long can tangle around the tool, thus causing injuries to operators and damage to

cutting tools, other than a poor surface quality of the workpieces. As a matter of fact, to avoid the above conditions, it is essential to intervene in the shortest time possible. To do so, this study proposes to combine the PdM strategy with the EC paradigm and demonstrates that it is possible to achieve impressive results in enabling PdM at the edge, anticipating equipment failures through an inference time in the order of milliseconds.

Our methodology adapts edge ML technique to address the specific challenges of the turning process. These challenges include the need for continuous monitoring, real-time analysis and timely intervention to prevent failures or inefficiencies. Turning, as well as other machining operations, can benefit significantly from edge ML, as these operations are intrinsically sensitive to faults, but with different specifications. For example, in milling, the movement of the tool relative to the material results in less uniform wear. In contrast, the continuous, high-speed nature of turning results in rapid and uniform wear. Therefore, in milling, continuous control of tool trajectories is necessary, while in turning, continuous control of cutting forces is required.

Executing real-time control through the Cloud Computing paradigm negatively affects the quality of monitoring critical variables. Cloud-based ML typically causes latency on the order of tens of milliseconds, which can render latency-sensitive applications unusable. In contrast, edge-based ML reduces latency to a few milliseconds, enabling the prediction of cutting tool wear and deterioration in real-time.

Therefore, with this time-saving solution, manufacturing companies address turning challenges proactively, avoiding unplanned and costly downtime and ensuring greater process safety. To demonstrate the effectiveness of the proposed approach, three devices were selected, i.e. three microcontrollers that differ mainly in terms of their computational capacity. Once an agnostic model has been defined with respect to the hardware used, the aim is to provide a comprehensive overview of the performance achievable on different hardware. The analysis conducted in this research revealed that the parameters of speed, power consumption and cost are not linearly dependent, and thus emphasizes the importance of a careful evaluation of the available hardware options for an optimized choice for specific application requirements.

The rest of this article is organized as follows. In Section II, the maintenance problem is presented. In Section III, we examine the related literature. Section IV discusses about the EC paradigm. In Section V, we present a turning problem and the solution we propose for it. Finally, Section VI concludes this article.

II. MAINTENANCE

A. MAINTENANCE HISTORY

Industrial and technological evolution has led to the identification of different techniques and strategies that can best adapt to the needs of modern industry. These needs require the optimization of industrial processes, which can only be achieved with highly automated systems. As industrial production has become more automated, the need for more efficient maintenance strategies has increased. The goal of these strategies is to improve the availability and efficiency of manufacturing processes. This can only be achieved by identifying any anomalies in real-time before the actual machine failure occurs, thus avoiding interrupting production and possibly causing costly unplanned downtime. Maintenance strategies, thus, evolved in order to better fit the aforementioned requirements.

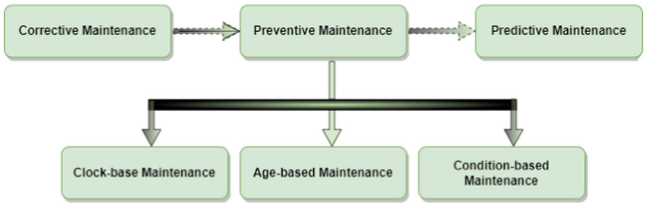

FIGURE 1. Types of maintenance strategies in manufacturing.

Overall, as shown in Fig. 1, three different maintenance strategies can be applied in the industrial field: 1) Corrective Maintenance; 2) Preventive Maintenance; and 3) Predictive Maintenance.

Corrective maintenance, also called failure-driven maintenance or RTF, is an unplanned maintenance type whose logic is "When a machine breaks down, fix it" [9]. It is the oldest and most common maintenance and repair strategy [10] which applies no-maintenance management before the breakdown of the machine occurs. The goal of Corrective maintenance is to bring the item back to a functioning state as soon as possible, either by repairing or replacing the failed item or by switching to a redundant item [11]. This maintenance strategy may apparently be the simplest to apply, as it is performed only when strictly necessary and it does not require any additional expenses to monitor the machinery during its life cycle. Furthermore, since the repair/replacement of a component takes place only when strictly necessary, the corrective maintenance strategy has the advantage of maximizing the time of use of the system. As a consequence, corrective maintenance can only be chosen if the failure of the machinery is not catastrophic since, in this case, the advantage of increasing the time of use of the system would have negative consequences, such as the increase in repair costs. In fact, the analyses on maintenance costs show that the repair in reactive mode has a cost generally three times higher than the cost of the same repair carried out with scheduled maintenance [9]. Moreover, as [12] points out, the machine uptime indicates the success of a maintenance strategy, therefore it is important to maximize this value. However, maximizing the machine uptime by delaying maintenance, as corrective maintenance does, may not be a winning strategy. In fact, this could lead to high machine downtime once the failure has occurred. For example, it is shown that, when applying a corrective maintenance strategy, high machine downtime is very frequent, since it is not certain

that the identification and resolution of the problem are immediate. Indeed, in the absence of other information, identifying the component that requires maintenance may take time.

Whether the (momentary or permanent) unavailability of machinery begins to prevail, it means that high and frequent machine downtime lead to reduced machine uptime, which is the opposite of the established goal.

The consideration presented above is a valid reason to switch from a corrective maintenance strategy to a type of maintenance that intervenes at predetermined intervals, defined as preventive maintenance [10].

Preventive maintenance, or TBM, is a maintenance strategy that schedules periodic checks in advance, in order to reduce the probability of system breakdown. Such periodic checks allow understanding when to repair or replace the system/component or even detect developing problems.

Differently from Corrective Maintenance, which is failure-driven, the frequency of Preventive Maintenance activities can be determined by the following factors.

1) *Clock-based maintenance:* Scheduled based on specified calendar times.
2) *Age-based maintenance:* Scheduled according to the specified age of the item.
3) *Condition-based maintenance:* Scheduled based on the value of one or more condition variables.

The underlying concept of those three maintenance strategies is that preventive maintenance strategy assumes that the system deterioration in normal usage is statistically or experimentally known. This means that it is possible to schedule a maintenance activity on a critical component at fixed intervals, just shorter than their lifetime.

However, if on one hand preventive maintenance can lead to organizing production stops in a more convenient way and can reduce the probability of failure, on the other hand, it cannot eliminate the occurrence of random catastrophic failures. Moreover, there is a risk of falling into excess or a defect of prevention, especially if it is not possible to estimate the RUL of a given component.

From this point of view, observing the failure curves of a certain machine is not very helpful. Indeed, failure curves do not consider the scenario in which a machine operates, when in fact the scenario has a direct effect on the life of a machine and its components. As suggested by several studies, the same item in different scenarios does not require the same maintenance schedule: In many cases, there is the risk of carrying out the replacement (prematurely) when it is not yet necessary. On the other hand, even avoiding reaching the end of life of a component, according to the bathtub curve (Fig. 2), the failure rate can still increase because the new equipment may experience infant mortality [13].

Moreover, [13] showed that time-based maintenance is unreliable even taking into account the working conditions of a specific tool. This consideration can be derived from the experiments done by the SKF Group, which performed a stress test on 30 identical bearing elements under identical conditions, to cause them to fail. In addition, it was observed

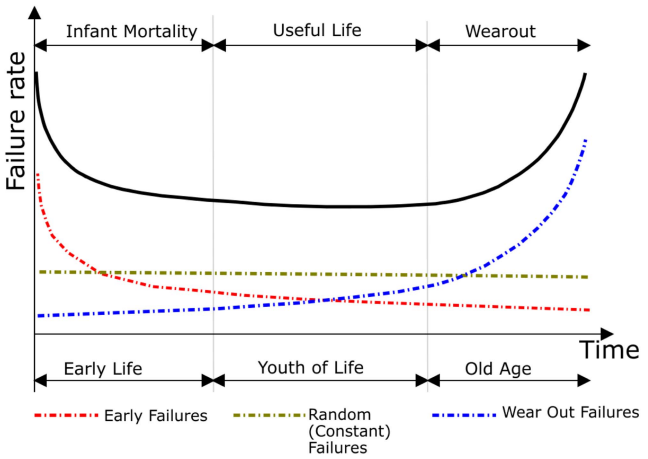

FIGURE 2. Bathtub curve.

that the time to failure of each component has a high variance despite the elements tested being all the same, as well as their working conditions [13].

Another aspect to take into consideration is the presence of spare parts, the management of which is linked to maintenance results. Unreliable maintenance can lead to an underestimation of the number of stocks needed. Thus, it may be hard to replace the faulty component in short time, with a consequent production stop which is equivalent to economic damage for the company.

When traditional maintenance strategies are adopted, engineers have to choose between maximizing the useful life of a system/component at the risk of machine downtime (unplanned maintenance) and maximizing uptime (time-based preventive maintenance) through early replacement of potentially good parts [14]. It has been shown that for many industrial companies, this type of organization is inefficient because, even if in theory the goal is to reduce machine downtime, in practice acceptable results are not achieved.

According to [5] as a result of the new policy, called PdM, it enables to lessen 25%–35% the maintenance costs, eliminate 70%–75% breakdowns, reduce 35%–45% breakdown time, and increase 25%–35% production. These results are a good reason to take a closer look at the most recent development in terms of PdM strategy.

B. PREDICTIVE MAINTENANCE (PDM)

The latest evolution of maintenance policies is called PdM which is a technique for optimizing the use interval of the equipment while reducing the frequency of maintenance activities to a minimum.

PdM could significantly reduce both maintenance activities and costs, thus reducing the waste of human and material resources. As a result, PdM helps avoid overmaintenance and undermaintenance while decreasing the risks of unexpected breakdowns. This result is achievable because maintenance

activities are scheduled based on the performance or conditions of the equipment and not based on a time-regular basis.

Despite the fact that predictive maintenance steps are not well defined, the key points of this strategy are known and will be classified hereafter.

One of the key points consists of spotting all the system critical variables. The aforementioned is a complex task that requires a deep knowledge of the monitored asset, its goals, and the environment effects on the monitored asset. Once these preconditions are met, it is possible to derive a model of the machining process, necessary to build a smart monitoring system.

Another key point of predictive maintenance strategy involves the monitoring of such variables. In fact, condition monitoring, i.e., the monitoring of critical variables, is the concept behind PdM. Depending on the peculiar working environment, there exist a lot of different technologies that can be used to collect data from the working environment, including both direct and indirect modes. During monitoring, the sensor signal has to be transformed into features that can adequately describe the signal. In particular, a continuous monitoring system provides a continuous flow of information, thanks to which a potential fault in the in-service equipment can be detected.

Finally, the last key point consists of the activity of prediction and decision-making. Indeed, data, obtained from the activity of monitoring the asset status, can be analyzed in order to predict the future behavior/condition of machines. This prediction allows a fault to be diagnosed even if the monitored machine has not failed. This is possible because a fault manifests itself as a deviation of the system or machine behavior from its nominal behavior. Therefore, we can identify on time a certain anomaly, i.e., a potential equipment fault, in order to make a decision before the degradation state develops into a worse one over time.

Different from the reactive maintenance technique, in which maintenance can be performed only after a failure, a proactive maintenance technique, such as PdM, requires maintenance to be performed before the occurrence of a failure. Actually, even adopting a proactive maintenance technique, it still seems the monitored asset fails suddenly. This is due to the fact that a machine component may enter a degradation state before it fails. However, if a degradation condition can be measured, it is possible to identify the component (or the components) which causes the failure and then remove it from the system early, which is exactly the main aim of proactive maintenance [15].

Once the fault is eventually detected, the next thing to do is to schedule a maintenance activity, before the detected fault develops into a worse degradation condition, eventually leading to the machine breakdown. Often undertaking a maintenance activity when a component has not failed yet could seem like a waste because that component is still able to accomplish its mission. Actually, it is important to point out that undertaking a maintenance activity when a component is just working in a degraded state (but still it is working) has the main benefit of preserving the high quality of the final product, which is one of the most important requirements of modern industry.

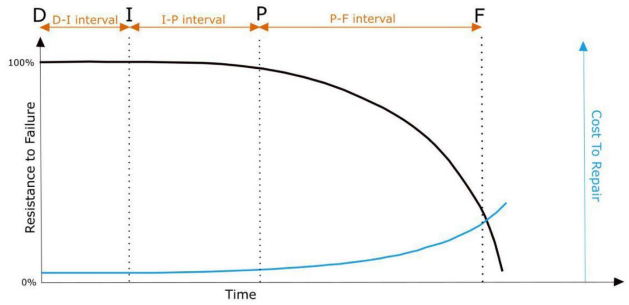

FIGURE 3. DIPF curve and CostToRepair curve.

The two main approaches that can be distinguished in PdM are the data-driven approach and the model-based approach.

The model-based approach has the ability to incorporate a physical understanding of the target product, relying on the mathematical model to represent the behavior of the system [16].

Instead, data-driven approaches (called ML approaches) are able to find highly complex and nonlinear patterns in data of different types and sources and transform raw data into features spaces, so-called models, which are then applied for prediction, detection, classification, regression, or forecasting [17]. This type of method does not require an in-depth understanding of system physical processes that lead to system failures and does not assume any underlying probability distributions [18], since it uses historical data to learn a model of system behavior [16].

Today, data-driven are the most popular approaches in fault diagnostics, since the availability of data is increasing.

To sum up all the benefits mentioned so far condition monitoring and ML are changing the way of performing maintenance activity, making it a more efficient, organized, and least cost action.

C. DIPF

The importance of taking real-time decisions can also be explained with the curve called DIPF, which is depicted in Fig. 3. The x-axis of the curve represents Time or Operating Age, while the y-axis represents Resistance to Failure.

Along this curve, it is possible to identify two important points for the predictive maintenance technique: the potential failure point (P) and the functional failure point (F).

P represents the potential failure point, i.e., it is physically possible to identify, through the monitoring of the parameters, a deviation of the asset behavior with respect to the condition of normal operation that indicates a future failure.

F is the functional failure point, that is to say the temporal instant in which the equipment is no longer able to run (or is not able to maintain a certain standard) and so maintenance is required.

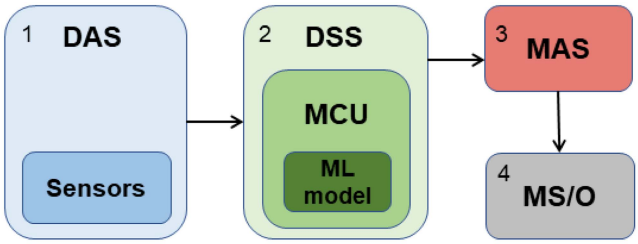

FIGURE 4. Block scheme of the proposed monitoring system.

The time elapsed between point P and point F is known as the P-F interval. Observe that during the P-F interval, the monitored asset has the ability to continue running even if it is in critical conditions. However, after the P-F interval, the asset is no longer able to complete its mission.

Once the point P has been identified, the time duration of the P-F interval is not unique but depends on the component: it can last months, weeks, or much less, as minutes or seconds. Knowing this interval duration is crucial to defining the best maintenance strategy to apply. Regardless of the case, the different maintenance strategies lead to lesser or greater rapidity of intervention.

In Fig. 3, the CostToRepair curve is also depicted in correspondence with the DIPF curve. The former curve increases increases exponentially in correspondence with point F (the closer we get to the breaking point F). Therefore successfully detecting potential failures requires intervening before the PF interval ends, so that the maintenance costs do not increase excessively. For the reasons explained above, PdM aims to predict the failure by detecting the problem early.

As stated by [19], to meet latency requirements, different architectures for quick-performing model inference have been proposed: 1) on-device computation, where machine learning algorithms are executed on the end device; 2) edge server-based architectures (where the data are sent from the end devices to edge servers for computation); and 3) joint computation which includes the possibility of having cloud processing.

D. FUNDAMENTAL BUILDING BLOCKS FOR AI-ENABLED PDM

To set up a PdM strategy, it is necessary to choose which approach to use to predict a failure event.

This article focuses on the application of the data-driven approach for PdM. As shown in Fig. 4, the AI-Based PdM approach is divided into the following blocks:

1) Data Acquisition System (DAS)
   At the basis of predictive analysis, it is necessary to determine which parameters must be continuously monitored. That is, to monitor the critical variables that influence the outcome of the process in order to detect a potential failure event. Sensor technologies, strategically placed in equipment and machines, form the core of the DAS. The functionality of this block ensures detailed data collection as it includes both historical and current records on the life of the equipment. This feature provides a temporal perspective of the system; hence, a complete understanding of the dynamic environment in which the PdM system is immersed.
2) Decision Support System (DSS)
   The DSS module integrates AI techniques. In particular, the Dataset provided by the DAS block allows the development of an ML algorithm, the most suitable for the given application. Then, based on previous experience, the predictive model learns to detect an anomaly or deviation in the data trend, returning a snapshot of the health of the equipment in the immediate future. As a result, the DSS block, the heart of the PdM strategy, predicts a potential failure with the aim of avoiding unexpected downtime in operations.
3) Maintenance Alert System (MAS)
   By observing the behavior of the machine, it is possible to identify whether it deviates from its nominal behavior. Consequently, a decision can be made before the degraded state of the equipment evolves into a worse one over time. In fact, based on the decision returned by the DSS block, the MAS block recommends a very good strategy, i.e., whether it is necessary to start a maintenance procedure on the machine before the physical system is irreversibly damaged.
4) Manufacturing System/Operators (MS/O)
   The last block in the decision-making chain is the MS/O whose task is to transform the possible maintenance decision of the MAS block into a concrete action. The preventive machine downtime procedure actively involves the operators working directly in the field. These receive trigger alerts and are responsible for interpreting the information, i.e. they have the task of scheduling when maintenance actions are required.

III. RELATED WORK

The most common way to tackle the problem of PdM is using ML algorithms whose goal is to increase the effectiveness of the maintenance activity. This article focuses on the application of ML models in the manufacturing industry, particularly the field of turning process.

Hereafter different solutions proposed in the literature are listed, each for a different sector. The researchers faced distinct problems, ranging from the prediction of cutting forces to the prediction of surface roughness, from the classification of the chip to the prediction of tool wear, etc. Each of them is particularized considering very precise working conditions: machine settings (such as the depth of the cut), type of material processed, and much more. In the literature there are some papers closely related to the task of chip form prediction, which is also the task tackled in this paper. In particular,

several of these deal with the choice of input data from which to infer the shape of the chip. There are several articles in the literature, [20], [21], [22], and [23], that use an ANN for their tasks.

In [20], the researchers focused on both the problems of chip form classification according to the standard ISO 3685-1977(E) and tangential cutting force prediction of cast nylon in turning operation. The final results showed an accuracy of 86.67% for classification chip form and accuracy of 91.130% for the main cutting force prediction. To train and validate the ANN, a small dataset was used (60 samples for training and 15 samples for validation). The cutting force was measured by a load cell, inserted under the cutting tool adaptor. The cutting force signal was sent to the computer via an interface card in order to collect the data.

In [21], the authors carried out a classification of the shape of the chip on the basis of the characteristics reported by the ISO 3685-1977 standard, which defines different classes in which to classify the shape of the chip. The latter (target of the ML model) is the result of the turning process, and it depends on the three components of the cutting force (which are the input data to the ML model). These input data are measured by means of a piezoelectric sensor incorporated in the tool holder.

The authors in [22] used emission signal analysis to predict the chip form during the cutting process. The prediction achieves a percentage of correct recognition of the chip form higher than 90%.

The article [23] performs a sensor monitoring of chip form in turning of C45 carbon steel through features from signals provided by a cutting force based sensor. In this sense, the pre-established goals are single chip form classification and favorable/unfavorable chip type prediction. As a result, through the NN, the obtained success rates in chip form recognition were always higher than 80%.

The authors in [24] focused on the control of chip formation during longitudinal turning of carbon steel with coated carbide inserts. The control for favorable chip formation has been carried out using real-time cutting force sensor signal spectrums. An unsupervised NN, specifically an SOM, was used to address this problem.

Another common problem addressed in literature consists in the prediction of the cutting forces in a turning process, which is a regression problem. The authors in [25] finds out a NN is more suitable for this task than a traditional linear regression model, since NN shows a greater accuracy.

Also, [26] showed how NN worked better than the various regression models in predicting the surface roughness, which is an indicator of surface quality, in the turning process in various conditions.

For example, Wenkler et al. [27] proposed a model for dealing with the problem of predicting the specific cutting force in a milling process. The model adopted to carry out this task is an ANN feedforward, which is a supervised model in which there is no back link between the neurons of one layer and those of the previous layer. Furthermore, the model is trained following an iterative process, as the milling process is continuously monitored and the specific cutting force is predicted from time to time. The cutting force F_c is not a directly adjustable or measurable parameter, therefore a further relationship is required. The input vector contains 13 parameters, the influencing parameters, which are each transferred as a scalar to the ANN.

The objective of [28], on the other hand, is to predict the surface roughness in the milling process. Three different models were compared, namely: RA, SVM, and BNN. The study examines how the surface roughness is influenced by the following parameters: The cutting speed, the cut itself, and the depth of cut. The result of the work shows that all three models, mentioned above, have a prediction error below 8%. In addition, when the size of the training dataset is small, the three models have comparable performance and the results are even better.

In [29], the authors used an ANN to monitor tool wear of milling operations. The authors carried out a classification of the tool status through an ANN model trained with data collected by acceleration sensors. This article also reports the comparison between different trained models and the result shows one greater efficiency of the ANN model compared to the SVM and KNN models. In addition, the work can be modified to predict the lifetime left to the tool using an ANN based on a supervised regression. In this case, it is necessary to acquire the acceleration data for a long time in order to progressively monitor the wear of the tool, from the beginning of the life cycle until the break.

In the study [30], the main research objective is to detect faults in bearings using a minimum set of observations and selecting the minimum number of features. The author applied vibration signals to predict deterioration. He uses vibration analysis to obtain features in an optimal ML model using a public dataset from Case Western Reserve University (CWRU), which contains data on bearing failures. As a result, the Kernel Naive Bayes model achieved an accuracy of 94.4%, while the Decision Tree (Fine Tree) and KNN, in detail Fine KNN, models demonstrate exceptional accuracy, achieving a perfect accuracy rate of 100%.

The authors in [31] presented a tool wear prediction model based on cutting forces measured in-process during peripheral milling of Ti-6Al-4V. It explains how the residual stress state of the machined subsurface influences the service quality indicators of a component. Indeed, during machining, the radius of the cutting edge changes due to tool wear. In specific, the rounding of the cutting edge significantly affects the residual stress state in the workpiece and the process forces occurring. An ML algorithm, a MLP model in this case, was implemented to calculate the effect of the change in cutting edge microgeometry due to tool wear using the radius of the cutting edge and the measured cutting forces.

In [32], a SVM model was used for implementing a tool breakage detection system in a milling process. The obtained result was compared with the traditional regression approach MVR. The cutting forces in input were measured during

the production process using a dynamometer. The proposed model can be recommended due to its slightly higher accuracy. However, it has been observed that the proposed model has to be trained through a relatively complicated tuning process of design parameters.

In [33], a maintenance activity was used to monitor the progression of lateral wear of the tool. In particular, a CNN model was chosen for its ability to find the correlation between the forces produced during the cutting processes and tool wear. The accuracy obtained is 90%.

In [34], the prediction of tool wear in milling operations was undertaken using three ML algorithms, such as ANNs, SVR, and RFs. The performance of these algorithms was evaluated on a dataset collected from 315 milling tests. A set of statistical features was extracted from cutting forces, vibrations, and acoustic emissions. The results have shown that while the training time on the particular dataset using RFs is longer than the FFBP ANNs with a single hidden layer and SVR, RFs generate more accurate predictions than the FFBP ANNs and SVR. But, as the authors themselves state, the ANN model employed has only a single hidden layer. This model may not be suitable for the considered task, since it is known that ANN better performs with more hidden layers.

The examined works are summarized in Table 1. Herein, it is easy to see that NN algorithms are applied to tackle different problems from each other, such as chip form prediction, prediction of the specific cutting force, surface roughness, and tool lifetime. In most of these problems, NNs have been demonstrated to be a competitive alternative to traditional classifiers for many practical classification problems [35].

Indeed, from the study conducted in [36] regarding the algorithms used for the prediction in the manufacturing sector, 28% of the problem of PdM in the manufacturing industry (which is also the highest fraction) is solved using an ANN model.

In this section, we only present works related to PdM since, to the best of our knowledge, there are no works combining PdM and EC coordinated to our field (turning process). The literature offers several reviews but no work explains the strategies introduced at a practical level and, therefore, in our field, there are currently no results to compare with.

TABLE 1. Summary of the Most Relevant Contributions in the PdM Together With ML in Related Works

Author	Target/process	Algorithm	Input
[20]	chip form classification and tangential cutting force/Turning	ANN	cutting force
[21]	chip form classification/Turning	ANN	cutting force
[22]	chip form classification/Turning	ANN	acoustic emission
[23]	chip form classification/Turning	ANN basedwork	cutting force
[24]	chip form classification/Turning	unsupervised NN	cutting force
[25]	cutting forces/turning	ANN	speed, depth of cut and feed rate
[26]	surface roughness/turning	ANN	depth of cut, feed rate, and insert nose radius
[27]	specific cutting force/milling	ANN	13 parameters
[28]	surface roughness/milling	RA,SVM and BNN	cutting speed, cut itself, depth of cut
[29]	tool wear/milling	ANN	acceleration
[30]	bearing	Kernel Naive Bayes, Decision Tree (Fine Tree), and KNN (Fine KNN)	vibration signals
[31]	peripheral milling	MLP	radius of the cutting edge and cutting forces
[32]	tool breakage detection/milling	SVR	cutting force
[33]	progression of lateral wear/milling	CNN	cutting force
[34]	tool wear/milling	ANNs,SVR, and RFs	cutting force, vibrations, acoustic emissions

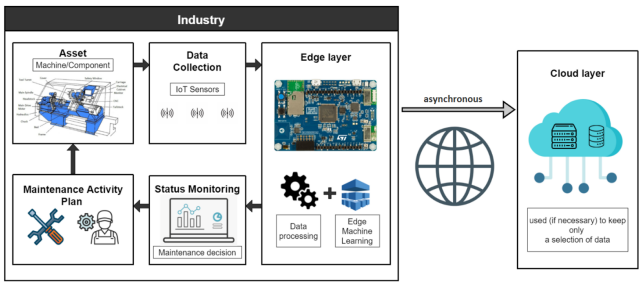

FIGURE 5. Edge computing paradigm.

IV. EDGE ML

EC (Fig. 5) is a paradigm that allows overcoming the limit of the cloud computing paradigm (Fig. 6).

Indeed, although the cloud is seen as "characterized by virtually unlimited capabilities in terms of storage and processing power" [37], a cloud-based approach has several disadvantages, as shown below.

For example, the increasing number of IoT devices leads to the generation of a large amount of data. Sending all this data toward the Cloud requires incredibly high network bandwidth [38]. This causes the Cloud to become a bottleneck, thus leading to a latency increase, and in general degrading the QoS.

Another consequence of the increased volume of data is the need for more storage capacity and computational capability. In some scenarios, meeting the aforementioned requirements may not be easily achievable by the Cloud service.

Moreover, there are many applications that require low latency (within a few milliseconds). Using Cloud services often means offloading data elaboration from the source to the Cloud. This procedure clearly involves the transfer of data from the source to the cloud and vice versa. The time needed to transfer data in the network may not be acceptable for some

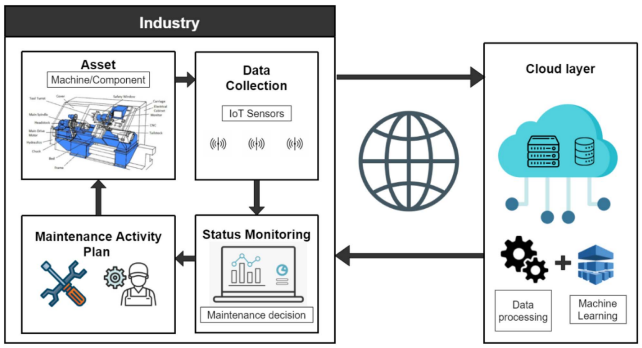

FIGURE 6. Cloud computing paradigm.

applications, since it depends on the geographical distance between the source and the cloud servers.

Moving to the Cloud also arises security and privacy concerns. Indeed, the cloud computing approach implies moving data from the source to third-party servers. Therefore, privacy and confidentiality issues arise due to the fact that an untrusted service provider necessarily has access to all the data, also considering that the physical location of data in the Cloud is unknown.

For the aboveexplained reasons, there are several applications where a cloud-based approach may make it difficult to comply with their real-time constraints. For example, in Healthcare Industry 4.0, patients' health data must be immediately available and any delay or failure introduced by the Cloud cannot be tolerated [39].

Similar real-time constraints are present in several applications such as industrial automation, virtual reality, real-time traffic monitoring, smart home, smart sea monitoring, data analytics, or maintenance, which is the case studied in this work.

For these types of applications, it is impractical and often it is not necessary to directly transfer raw data to a remote cloud [40]. An alternative solution could be the adoption of the EC paradigm that allows data to be processed locally, as close as possible to the data sources, using smart devices.

In general, the EC approach can be defined as an extension of cloud computing [41] since both their structures are similar, but the main difference is in the positioning of the computing applications, data, and services, which are no more located in central nodes, the core, but in the other logical extreme, the edge of the Internet [42].

To be precise, we have also to point out that the main disadvantage of EC over Cloud Computing is that edge devices have limited computational power and storage capacity. However, it is still possible to carefully choose the edge devices by ensuring they have features like large enough memory, sufficient process capabilities, and sufficient Internet bandwidth, which allows connecting with the cloud if it is necessary.

Consider, e.g., our case study in which, in order to monitor an asset condition, a large amount of data must be collected by real-time sensors. This data must, then, be processed (adopting a machine learning algorithm) in order to obtain information on the current state of the monitoring asset and make predictions about its future state.

In addition, in the event that the DIPF curve of the asset is characterized by a very short P-F interval (of the order of seconds), it is impossible for a human being to encounter the potential failure and intervene before the functional failure. Therefore in this case it is crucial to opt for strategies able to reduce as much as possible the time needed for detecting a potential failure and making a subsequent decision on how to treat that failure. Opting for Edge ML strategies is therefore a must in these cases.

Adopting edge devices to process the collected data has several benefits, mainly due to the proximity of the data source to these edge devices, such as low transfer latency, context and location awareness, high scalability and availability, all while maintaining confidentiality of possibly sensitive data.

Therefore, edge devices enable real-time computation, which is fundamental to determining whether the current manufacturing equipment state is or not normal. In case of anomalies, it enables to take real-time decisions like blocking the productive process, either by sending an alarm to the employee, or executing an interrupt routine that it has previously learned.

V. CASE STUDY

The scope of the research work is to monitor the longitudinal turning process of carbon steel, with metal-coated inserts taking under control the chip form classification.

The normal variations of process conditions can produce changes in the chip form or shape during a machining operation.

The problem with the formation of chips that are too long is that it complicates, in many cases, the machining process, as a too long chip may tangle around the tool.

Unacceptable chip shapes can cause injuries to operators and damage to cutting tools, workpieces (resulting in a decrease in surface quality) and machines [44], [45].

In this work, a monitoring system of cutting force sensor signals is proposed to predict the shape of the chip through a scaled ML algorithm for application on an end-device.

Given these considerations, this turns into a PdM problem, as we want to monitor the parameters that affect chip formation in order to detect, and possibly interrupt, a potentially dangerous process.

For the above problem we consider three chip shapes, defined by the ISO 3685 standard [46] (as shown in Fig. 8): 1) short spiral; 2) short; and 3) snarled.

The dataset was obtained by monitoring the three components (F_c, F_f, F_p), represented in Fig. 7, of the cutting force measured using a Kistler laboratory dynamometer 9263 [47], a three-channel piezoelectric dynamometer.

The cutting parameters set to obtain the cutting forces, are as follows:

1) *Cutting speed* = 150, 250 m/min;

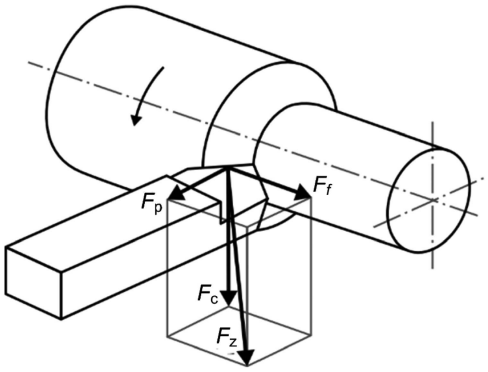

FIGURE 7. Cutting forces generated during the turning process (Figure taken from [43]).

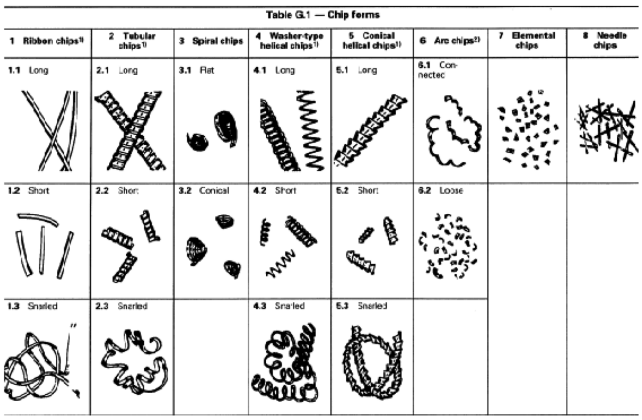

FIGURE 8. ISO standard 3685.

2) *Feed* = 0.08, 0.13, 0.20, 0.30 mm/rev;
3) *Depth of cut* = 1.0, 1.5, 2.0, 3.0 mm.

The force component signals were digitized at a frequency of 2500 Hz for three seconds, resulting in a data sequence of 7500 points. Then, an analysis of cutting force signal specimens is performed.

The following steps of our methodology can be defined.

1) Cutting force signal specimens are measured with three-channel piezoelectric dynamometer.
2) LPA, in particular Durbin's algorithm, is applied to estimate the spectral characteristics of the signal.
3) The characteristic predictive coefficients of the spectral model are derived, four features for each cutting force.

The analysis of CFS specimens is carried out by achieving spectral estimation through a parametric method. In this procedure, the signal spectrum is assumed to take on a specific functional form, the parameters of which are unknown. The spectral estimation problem, therefore, becomes the estimation of these unknown parameters of the spectrum model rather than the spectrum itself. From the (measurement vector), p features or predictor coefficients (feature vector), $\{a_1, \ldots, a_p\}$, characteristic of the spectrum model, are obtained through LPA. Feature extraction is executed through

TABLE 2. F_c Component

#n	a1	a2	a3	a4	Label
1	0,8669359	0,04649681	0,06447785	0,02127867	Snarled
2	0,8792428	-0,01853898	0,1564354	-0,01799053	Short
...	...	...	...	...	...
210	1,017999	-0,2595151	0,1090982	0,1314222	Short Spiral

TABLE 3. F_p Component

#n	a1	a2	a3	a4	Label
1	0,6668675	-0,2010417	0,5442709	-0,01123788	Snarled
2	0,6806869	-0,2257641	0,5668689	-0,02312554	Short
...	...	...	...	...	...
210	0,9141559	-0,3159226	0,3352912	0,06163793	Short Spiral

TABLE 4. F_f Component

#n	a1	a2	a3	a4	Label
1	0,6234717	0,2742358	0,07645185	0,02476593	Snarled
2	0,8600896	-0,08066341	0,2481833	-0,02873434	Short
...	...	...	...	...	...
210	1,047113	-0,5930006	0,6296181	-0,08627542	Short Spiral

TABLE 5. Dataset Summary

	#n observation	Label
class 0	45	Snarled
class 1	90	Short
class 2	75	Short Spiral
total	210	

the application of Durbin algorithm and the p value is chosen by examining the plot of the normalized RMS prediction error versus the order of the model [44].

The acquisition of the force sensors, the saving on a circular array and the real-time processing to extract the features of interest are compatible with the computational capabilities of many microcontrollers.

The above process is applied to obtain the experimental dataset. The dataset consists of the three cutting force components (F_c, F_f, F_p), each represented through four features ($a1$, $a2$, $a3$, $a4$). Table 2 –4, one for each component, show the values of the features for a part of the dataset. Overall this leads to 12 features plus a target. The target can assume 3 integer values: the value 0 represents the snarled shape of the chip, the value 1 indicates the short shape, and the value 2 represents the short spiral shape. The experimental dataset is composed of 210 observations; 45 for class 0, 90 for class 1, and 75 for class 3. For clarity, detailed information regarding the distribution of the dataset is summarized in Table 5.

The dataset is unbalanced, but not so much as to negatively affect the result. Therefore, it is not necessary to perform any operation of balancing.

In this work, we did not implement the data acquisition setup as we used the data collected in the paper [44].

A. SECOND STEP: MATLAB/PYTHON DATA ANALYSIS AND MODEL DESIGN

In this section, we focus on the choice of multiclass supervised classification models that, starting from the value of

the cutting forces, perform better in predicting the chip shape produced by the turning process.

MATLAB: To select the best models, we use the *Classification Learner App*, included in the Statistics and Machine Learning Toolbox for MATLAB R2021. The software can be used to train supervised and semisupervised learning algorithms for binary and multiclass problems [48]. We used this application to train the ML models (on the same dataset) belonging to the following classes.

1) Decision Trees.
2) Discriminant Analysis.
3) SVMs.
4) Nearest Neighbors.
5) Naive Bayes.
6) Ensemble and NN.

Each of these classes contains one or more models for a total of 29 trained models.

The ML models were trained without using the PCA. *Classification Learner App* offers three possible choices for the model validation: *K-Fold Cross Validation* (K-FCV), *Hold-out*, and *No Validation*. Among these, we choose to use the K-FCV method, setting the number of folds equal to 5. This method leads to better prediction accuracy, as it allows to avoid overfitting issues. Once the application trains each model, it reports the details in the "Summary" section. Also, the application presents the "Optimizer" option, which allows choosing the search range of the hyperparameters.

An example of the hyperparameter ranges used, in the case of NNs, is given below. These ranges were initially informed by established practices and empirical evidence in the field. Subsequently, they were fine-tuned to ensure optimal performance for our specific application. The intervals are as follows.

1) *Number of fully connected layers:* The maximum number of fully connected layers was set at 4. This range was determined on the basis of a balance between the complexity of the model and the computational burden required by deeper networks.
2) *Layer size:* The maximum number of neurons per layer was set to 100. This range was chosen on the basis of empirical data and the literature review, which suggest that this size is sufficient to capture the complexity of the data without excessively increasing the computational load.
3) *Activation function:* ReLU, which is effective in training deep neural networks, was chosen as the activation function.
4) *Maximum number of iterations:* The training process can perform a maximum of 100 iterations. This parameter was chosen in order to guarantee a sufficient number of training epochs while avoiding the problem of overfitting.

The obtained results are reported in Table 6. Herein, for each model, we specify: Classifier family, Classifier type, Accuracy (%), and Training time (Seconds). The results show different models predict with very high accuracy: Quadratic Discriminant Analysis, some types of SVM and KNN, Ensemble KNN, and NN manage to achieve accuracy between 95% and 99%.

TABLE 6. MATLAB Results

Classifier	Classifier type	Acc(%)	Prediction speed (Object/ Seconds)
Decision Trees	Fine Tree	92.9%	∼ 13000
	Medium Tree	92.9%	∼ 14000
	Coarse Tree	77.6%	∼ 11000
Discriminant Analysis	Linear Discriminant	74.8%	∼ 10000
	Quadratic Discriminant	99.0%	∼ 7600
Support Vector Machines (SVM)	Linear	73.8%	∼ 6000
	Quadratic	98.1%	∼ 5700
	Cubic	98.6%	∼ 7700
	Fine Gaussian	98.6%	∼ 6300
	Medium Gaussian	99.5%	∼ 5200
	Coarse Gaussian	74.3%	∼ 6300
Nearest Neighbors	Fine KNN	99.0%	∼ 6800
	Medium KNN	98.6%	∼ 7300
	Coarse KNN	51.9%	∼ 7700
	Cosine KNN	99.5%	∼ 6500
	Cubic KNN	97.6%	∼ 5400
	Weighted KNN	99.0%	∼ 6600
Naïve Bayes	Gaussian	69.5%	∼ 11000
	Kernel	84.3%	∼ 1700
Ensemble	Boosted Trees	42.9%	∼ 16000
	Bagged Trees	98.1%	∼ 2300
	Subspace Discriminant	71.9%	∼ 1400
	Subspace KNN	99.5%	∼ 760
	RUS Boosted Trees	96.7%	∼ 1800
Neural Network (NN)	Narrow NN	96.7%	∼ 14000
	Medium NN	98.1%	∼ 15000
	Wide NN	97.6%	∼ 10000
	Bilayered NN	98.1%	∼ 12000
	Trilayered NN	95.1%	∼ 11000

Actually, taking into account both the accuracy and the prediction speed, our results show that NNs present the best tradeoff between accuracy (ranging from 95% and 99%) and prediction speed (ranging from 10 000 to 15 000).

In addition, as highlighted in Section III, NN models represent the most used models in the field of Predictive Maintenance, via the ML approach. This is due to the fact that NNs show high robustness and they return a very high accuracy in different scenarios, such as predicting chip shape, predicting specific cutting force, surface roughness and tool life, etc.

We focused our attention on classification models that were suitable of guaranteeing adequate performance. Exploiting NN's models, an excellent tradeoff was chosen between the complexity of the problem addressed and the computational resources required. In particular, the intrinsic simplicity of the input data structure made it possible to avoid complex models that, although potentially more accurate, would have required significantly greater computational resources. In fact, a lighter model not only optimizes the use of the microcontroller's memory, but also reduces energy consumption, an important factor for embedded applications where energy efficiency is often a priority. The choice falls on NN models, in addition to

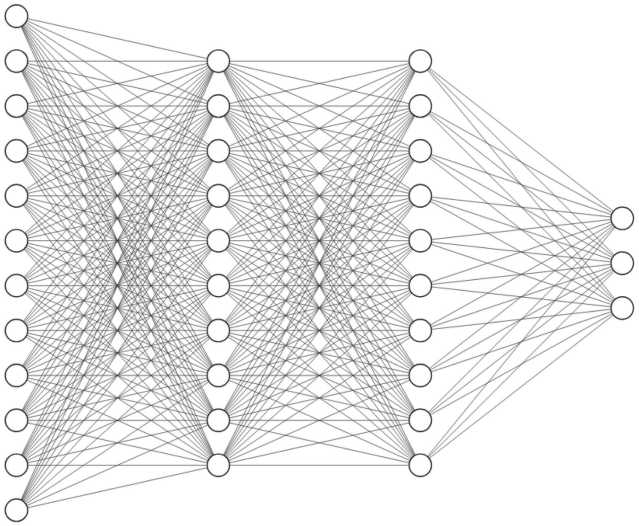

FIGURE 9. Structure of the Bilayered NN.

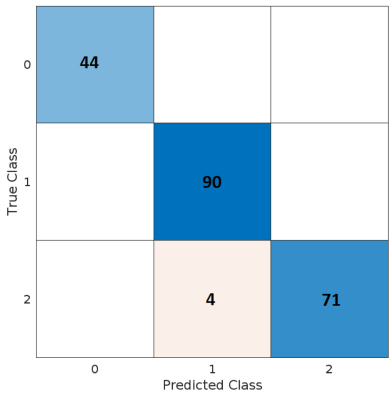

FIGURE 10. Validation confusion matrix for the Bilayered NN.

TABLE 7. Number of Neurons

Model	First Hidden Layer	Second Hidden Layer	Third Hidden Layer
1NN	100	-	-
2NN	100	80	-
3NN	20	100	50
4NN	100	100	100

TABLE 8. Desktop Results

Classifier type	Accuracy(%)	Training Time (s)
1NN	95.24	42.421
2NN	99.40	22.98
3NN	99.40	9.918
4NN	98.20	6.493

TABLE 9. Values of Complexity (MACC), RAM, and FLASH Memory Associated to Each Model

Model	RAM (KiB)	Flash (KiB)	Complexity (MACC)
1NN	1.86	16.62	1748
2NN	2.64	48.51	9848
3NN	3.01	41.01	7778
4NN	3.2	96.63	22148

the reasons already listed, also because state-of-art technologies are considered. The actual software technology primarily supports the deployment of NN models in microcontrollers. In fact, STM32 software focuses on optimizing NN models for efficient execution on embedded systems, offering, for these models, functionalities such as quantization and memory usage optimization [49].

Therefore, the selection of suitable models was limited to an efficient classification structure that could provide proper accuracy in a reasonable time while occupying a small amount of memory.

As reported in Table 6, 5 NN types were trained by the MATLAB Toolbox: Narrow NN; Medium NN; Wide NN; Bilayer NN, and Trilayer NN. The following settings were specified in the MATLAB Toolbox to train the NN models:

1) the activation function used for all the fully connected layers (except the last one) is *ReLU*;
2) the activation function used for the last layer is *softmax*;
3) the maximum number of training iterations is set to 1000;
4) the *Regularization strength (Lambda)* is set to 0;
5) data standardization is performed before the training process.

Classification Learner App allows visualizing the performances of the trained ML models. In the following, we report the Validation Confusion Matrix (Fig. 10) and the Structure (Fig. 9) for the Bilayered NN model.

Python: Each ML model was developed using Python on a Desktop environment. Following the preliminary analysis results provided by MATLAB Toolbox, NN models were implemented leveraging the TensorFlow library. In this way the model obtained is smaller or comparable to the memory available in the device used in this work.

Table 7 shows the architectures of the implemented models and Table 8 shows the accuracy percentage and training times (expressed in seconds) of the trained NNs.

B. THIRD STEP: MICROCONTROLLER IMPLEMENTATION

The performance of the algorithms was tested on different embedded boards from the STMicroelectronics family.

The devices selected were: NUCLEO-H743ZI2; B-U585I-IOT02A and NUCLEO-F401RE. Each board has a maximum frequency of 480 MHz, 160 MHz, and 84 MHz, respectively. The STM32CubeIDE tool with the Expansion Package X-CUBE-AI was employed to load the pretrained NN algorithms onto the microcontroller.

In detail, the models implemented with Python language were saved in *.h5* format, one of the formats supported by the Additional Software X-CUBE-AI. In this way, there is no need to use a separate transpiler, because the tool provides a native solution for converting TensorFlow models to C/C++, optimizing performance for the selected STM32 microcontroller.

Table 9 shows the four models with their relative complexity (MACC), the amount of RAM and FLASH memory

TABLE 10. Microcontroller Results

Model	Board	Time-to-inference(ms)
1NN	NUCLEO-H743ZI2	0.026
	B-U585I-IOT02A	0.103
	NUCLEO-F401RE	0.224
2NN	NUCLEO-H743ZI2	0.130
	B-U585I-IOT02A	0.453
	NUCLEO-F401RE	0.971
3NN	NUCLEO-H743ZI2	0.107
	B-U585I-IOT02A	0.370
	NUCLEO-F401RE	0.787
4NN	NUCLEO-H743ZI2	0.289
	B-U585I-IOT02A	0.984
	NUCLEO-F401RE	2.101

used. As can be seen from the table, the size of each model is such that there is no need to reduce the size of the model by compression or simplification. The peripherals used in this project encompass the Timer, necessary to calculate the inference time, and the Universal Synchronous Asynchronous Receiver/Transmitter for communication between PC and microcontroller.

Table 10 shows the measurement of the inference time, expressed in milliseconds, for each board.

The inference time indicates the time required to obtain the prediction output, that has a three-words format: Each value indicates the probability of belonging to the three classification classes. The obtained result confirms the choice of EC: The inference time is of the same order of magnitude as the delay in sending and receiving data from the cloud network. For our application, the delay that would accumulate using the Cloud infrastructure would be unacceptable. This means that a successful PdM strategy requires the implementation of EC. In fact, for delay-sensitive applications, it becomes imperative to allocate AI algorithms and data processing at the edge, in the proximity of the sensors, leveraging edge intelligence. Nevertheless, microcontrollers easily interface with standard industrial equipments, exploiting industrial buses and connection (e.g., SCADA, CAN, RS485) to real-time interact with MS/O. This allows the reduction of equipment off-time and the side effects of wasting material resources due to the derive of the quality production, influenced from potentially dangerous processes, e.g., the formation of snarled chips.

At the intersection of PdM and EdgeML, it is essential to emphasize the efficiency of industrial systems. Therefore, a comparative analysis is carried out based on the computational efficiency of the models, in order to evaluate the following:

1) Energy consumed during the inference operation.
2) Energy efficiency (MACC per Joule).
3) RAM and Flash memory utilization versus complexity.

1) Once the inference times are obtained, for each device used, it is possible to calculate the value of the energy consumed during the inference operation of ML models. In particular, the formula for energy consumption (in Joules) for electronic devices is given by

$$E = V \times I \times t \tag{1}$$

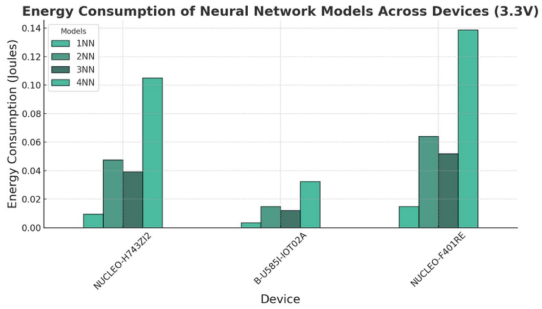

FIGURE 11. Energy consumption of NN models across devices.

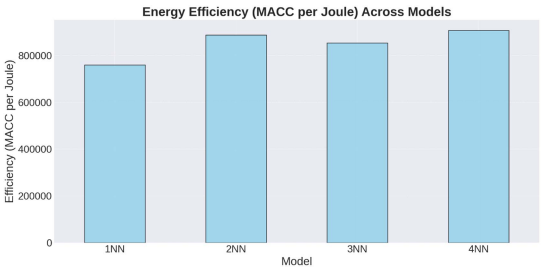

FIGURE 12. Energy efficiency (MACC per Joule) across models.

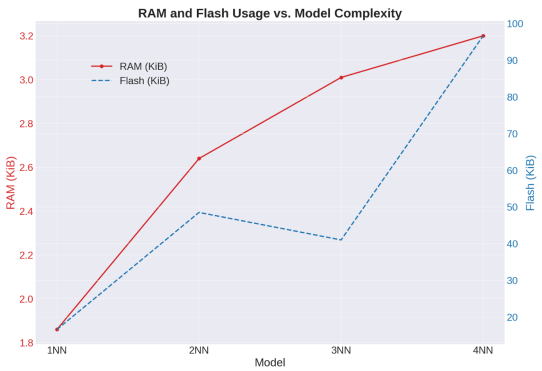

FIGURE 13. RAM and flash usage versus model complexity.

where E is energy, V is voltage, I is current, and t is time. The results obtained from (1) are shown in Fig. 11.

2) Energy Efficiency is defined as the number of multiplication-accumulation operations (MACC) that can be performed for each Joule of energy consumed. The results of this comparison are shown in the graph 12.

3) Fig. 13 correlates RAM and Flash memory usage with the complexity of each model. It is evident how the demand for resources is directly proportional to the complexity of the model.

Our analysis shows that the microcontroller in the midrange (intermediate frequency) outperforms the other two devices in terms of energy consumption. This counterintuitive result stems from the inherent differences between microcontrollers. In detail, the high-speed microcontroller benefits from advanced circuitry that enables it to perform its tasks faster,

while performing them at a higher current level. In contrast, the low-speed microcontroller requires a longer execution time to complete its tasks, resulting in higher power consumption over a longer period.

Together, these analyses offer a comprehensive view of how model complexity affects not only the computational efficiency, but also the resource requirements of NN models. Indeed, factors such as required computing power, energy consumption, and available memory resources can guide the choice of models and hardware for specific applications, especially in contexts where these factors are constrained. In this context, the search for a PdM strategy for industrial systems requires the maximization of efficiency, thus pushing towards solutions that optimize the cost of running industrial production processes. All this results in tradeoffs between the complexity of the model and the available resources, during the design phase of efficient and sustainable industrial systems.

VI. CONCLUSION

In the Industry 4.0 evolution, many limitations in standard maintenance approaches in terms of quality, efficiency, and latency have emerged, requiring increasingly high-performance strategies. PdM activities combined with edge intelligence represent a new trend that is transforming the manufacturing industry into an intelligent unit. The EdgeAI is implemented through smart devices capable of running ML algorithms and (pre and post) process data. In this way, task execution can be shifted, partially or entirely, from the cloud closer to the IoT sensors.

This paradigm enables decision-making and monitoring activities at the edge, especially for applications that require low latency. The transformation to smart manufacturing is already underway and growing rapidly. Nowadays, more research focuses on finding solutions to challenges and opportunities by exploiting the high potential of this method.

Our research lies in the abovementioned context. To the best of our knowledge, this is the first contribution that focused on developing an edge artificial intelligence to predict the class of the chip produced during a turning process.

This article introduced a methodology to select models that best fit the particular work environment. Moreover, being our application a delay-sensitive one, our research also focuses on the efficiency of the prediction process. From our study, it results that NN models offer the best tradeoff between accuracy and prediction speed. The trained NN models show an accuracy ranging from 96% to 98%. Moreover, our estimates show that the time needed to make an inference, employing an NN model, is in the order of a few milliseconds.

This promising result justifies the adoption of the EC paradigm in place of the cloud computing one. Indeed, the latter would require a higher prediction time because of the latency due to the time to transfer data over the network. Such time is at least of the same order of magnitude as the measured prediction time. This additional delay is unacceptable for most delay-sensitive applications, that require timely interventions to prevent damages to equipment and working materials. Moreover, the proposed solution leverages the resilience of the manufacturing process that become independent of the availability of network connection. In conclusion, the company's choice of a maintenance strategy is critical to improving the chance of success. Each company is responsible for the design of the maintenance technique in order to obtain one that perfectly fits its needs.

An important aspect to deal with in future works is the extension of the methodology to the long-term model lifecycle. In this context, the automation of model retraining and continuous performance monitoring represents a significant challenge for most real-world applications, as the simple initial training of models is not sufficient to guarantee consistent performance over time. Several factors, such as model drift, aging, continuous data collection, and management, as well as data security during retraining, can negatively affect the accuracy and effectiveness of ML models, especially in dynamic and constantly changing environments. What is referred to in the literature as the MLOps paradigm addresses these issues by providing a set of techniques that enable continuous monitoring of model performance over time. This involves the periodic evaluation of data, model, infrastructure resources, and model performance to detect potential errors or changes that may affect the quality of the product, keeping the model fresh and accurate.

REFERENCES

[1] E. Hozdić, "Smart factory for industry 4.0: A review," *Int. J. Modern Manuf. Technol.*, vol. 7, no. 1, pp. 28–35, 2015.

[2] J. Lee, "Smart factory systems," *Informatik-Spektrum*, vol. 38, no. 3, pp. 230–235, 2015.

[3] J. Lee, B. Bagheri, and H.-A. Kao, "A cyber-physical systems architecture for Industry 4.0-based manufacturing systems," *Manuf. Lett.*, vol. 3, pp. 18–23, 2015.

[4] J. Yan, Y. Meng, L. Lu, and L. Li, "Industrial Big Data in an Industry 4.0 environment: Challenges, schemes, and applications for predictive maintenance," *IEEE Access*, vol. 5, pp. 23484–23491, 2017.

[5] T. T. Van, L. H. Chan, S. Parthasarathi, C. P. Lim, and Y. Q. Chua, "IoT and machine learning enable predictive maintenance for manufacturing systems: A use-case of laser welding machine implementation," in *Proc. 12th Conf. Learn. Factories*, Apr. 3, 2022, pp. 1–6.

[6] M. M. Mabkhot, A. M. Al-Ahmari, B. Salah, and H. Alkhalefah, "Requirements of the smart factory system: A survey and perspective," *Machines*, vol. 6, no. 2, 2018, Art. no. 23.

[7] R. Carotenuto, M. Merenda, F. G. D. Corte, and D. Iero, "Online black-box modeling for the IoT digital twins through machine learning," *IEEE Access*, vol. 11, pp. 48158–48168, 2023.

[8] A. Lazzaro, D. M. D'Addona, and M. Merenda, "Comparison of machine learning models for predictive maintenance applications," in *Proc. Int. Conf. Syst.- Integr. Intell.* Springer, 2022, pp. 657–666.

[9] R. K. Mobley, *An Introduction to Predictive Maintenance*. Amsterdam, Netherlands: Elsevier, 2002.

[10] R. Kothamasu, S. H. Huang, and W. H. VerDuin, "System health monitoring and prognostics—A review of current paradigms and practices," *Int. J. Adv. Manuf. Technol.*, vol. 28, pp. 1012–1024, 2006.

[11] M. Rausand and A. Hoyland, *System Reliability Theory: Models, Statistical Methods, and Applications*, vol. 396. Hoboken, NJ, USA: Wiley, 2003.

[12] J. M. Gross, *Fundamentals of Preventive Maintenance*. New York, NY, USA: AMACOM/American Management Association, 2002.

[13] H. M. Hashemian, "State-of-the-art predictive maintenance techniques," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 1, pp. 226–236, Jan. 2011.
[14] C. Coleman, S. Damofaran, and E. Deuel, "Predictive maintenance and the smart factory," Deloitte Consulting LLP, 2017.
[15] J. Lee, "Machine performance monitoring and proactive maintenance in computer-integrated manufacturing: Review and perspective," *Int. J. Comput. Integr. Manuf.*, vol. 8, no. 5, pp. 370–380, 1995.
[16] M. Paolanti, L. Romeo, A. Felicetti, A. Mancini, E. Frontoni, and J. Loncarski, "Machine learning approach for predictive maintenance in Industry 4.0," in *Proc. 14th IEEE/ASME Int. Conf. Mechatronic Embedded Syst. Appl.* IEEE, 2018, pp. 1–6.
[17] T. Wuest, D. Weimer, C. Irgens, and K.-D. Thoben, "Machine learning in manufacturing: Advantages, challenges, and applications," *Prod. Manuf. Res.*, vol. 4, no. 1, pp. 23–45, 2016.
[18] D. Wu, C. Jennings, J. Terpenny, and S. Kumara, "Cloud-based machine learning for predictive analytics: Tool wear prediction in milling," in *Proc. IEEE Int. Conf. Big Data*, 2016, pp. 2062–2069.
[19] M. Merenda, C. Porcaro, and D. Iero, "Edge machine learning for ai-enabled IoT devices: A review," *Sensors*, vol. 20, no. 9, 2020, Art. no. 2533.
[20] B. Suksawat, "Chip form classification and main cutting force prediction of cast nylon in turning operation using artificial neural network," in *Proc. ICCAS 2010*. IEEE, 2010, pp. 172–175.
[21] Z. J. Viharos, S. Markos, and C. Szekeres, "Ann-based chip-form classification in turning," in *Proc. 17th IMEKO World Congr.–Metrology 3rd Millennium*, 2003, pp. 1469–1473.
[22] S. Sukvittayawong and I. InaSaki, "Identitication of chip form in turning process," *JSME Int. J. Ser. 3, Vib., Control Eng., Eng. Ind.*, vol. 34, no. 4, pp. 553–560, 1991.
[23] T. Segreto, A. Simeone, and R. Teti, "Chip form classification in carbon steel turning through cutting force measurement and principal component analysis," *Procedia Cirp*, vol. 2, pp. 49–54, 2012.
[24] A. Keshari et al., "Subtraction of inconsistence sensor data to improve the chip form classification and monitoring efficiency," in *Proc. 6th CIRP Int. Conf. Intell. Comput. Manuf. Eng.-CIRP ICME*, vol. 8, 2008, Paper 205.
[25] M. Hanief, M. Wani, and M. S. Charoo, "Modeling and prediction of cutting forces during the turning of red brass (C23000) using ANN and regression analysis," *Eng. Sci. Technol., Int. J.*, vol. 20, pp. 1220–1226, 2016, doi: 10.1016/j.jestch.2016.10.019.
[26] M. Nalbant, G. Hasan, and I. Toktas, "Comparison of regression and artificial neural network models for surface roughness prediction with the cutting parameters in CNC turning," *Modelling Simul. Eng.*, vol. 2007, 2007, Art. no. 092717, doi: 10.1155/2007/92717.
[27] E. Wenkler, F. Arnold, A. Hänel, A. Nestler, and A. Brosius, "Intelligent characteristic value determination for cutting processes based on machine learning," *Procedia CIRP*, vol. 79, pp. 9–14, 2019.
[28] B. Lela, D. Bajić, and S. Jozić, "Regression analysis, support vector machines, and Bayesian neural network approaches to modeling surface roughness in face milling," *Int. J. Adv. Manuf. Technol.*, vol. 42, pp. 1082–1088, 2009.
[29] D. F. Hesser and B. Markert, "Tool wear monitoring of a retrofitted CNC milling machine using artificial neural networks," *Manuf. Lett.*, vol. 19, pp. 1–4, 2019.
[30] M. Alonso-González, V. G. Díaz, B. L. Pérez, B. C. P. G-Bustelo, and J. P. Anzola, "Bearing fault diagnosis with envelope analysis and machine learning approaches using CWRU dataset," *IEEE Access*, vol. 11, pp. 57796–57805, 2023.
[31] M. Wimmer, R. Hartl, and M. F. Zaeh, "Determination of the cutting-edge microgeometry based on process forces during peripheral milling of Ti-6Al-4V using machine learning," *J. Manuf. Mater. Process.*, vol. 7, no. 3, 2023, Art. no. 100.
[32] S. Cho, S. Asfour, A. Onar, and N. Kaundinya, "Tool breakage detection using support vector machine learning in a milling process," *Int. J. Mach. Tools Manufacture*, vol. 45, no. 3, pp. 241–249, 2005.
[33] A. Gouarir, G. Martínez-Arellano, G. Terrazas, P. Benardos, and S. Ratchev, "In-process tool wear prediction system based on machine learning techniques and force analysis," *Procedia CIRP*, vol. 77, pp. 501–504, 2018.
[34] D. Wu, C. Jennings, J. Terpenny, R. X. Gao, and S. Kumara, "A comparative study on machine learning algorithms for smart manufacturing: Tool wear prediction using random forests," *J. Manuf. Sci. Eng.*, vol. 139, no. 7, 2017, Art. no. 071018.
[35] G. P. Zhang, "Neural networks for classification: A survey," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 30, no. 4, pp. 451–462, Nov. 2000.
[36] A. Du Preez and G. A. Oosthuizen, "Machine learning in cutting processes as enabler for smart sustainable manufacturing," *Procedia Manuf.*, vol. 33, pp. 810–817, 2019.
[37] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and Internet of Things: A survey," *Future Gener. Comput. Syst.*, vol. 56, pp. 684–700, 2016.
[38] M. De Donno, K. Tange, and N. Dragoni, "Foundations and evolution of modern computing paradigms: Cloud, IoT, edge, and fog," *IEEE Access*, vol. 7, pp. 150936–150948, 2019.
[39] P. Pace, G. Aloi, R. Gravina, G. Caliciuri, G. Fortino, and A. Liotta, "An edge-based architecture to support efficient applications for healthcare Industry 4.0," *IEEE Trans. Ind. Informat.*, vol. 15, no. 1, pp. 481–489, Jan. 2019.
[40] S. Wang et al., "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," in *Proc. IEEE INFOCOM 2018-IEEE Conf. Comput. Commun.*, 2018, pp. 63–71.
[41] W. Khan, E. Ahmed, I. Sahib, I. Yaqoob, and A. Ahmed, "Edge computing: A survey," *Future Gener. Comput. Syst.*, vol. 97, pp. 219–235, 2018, doi: 10.1016/j.future.2019.02.050.
[42] P. G. Lopez et al., "Edge-centric computing: Vision and challenges," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, pp. 37–42, 2015.
[43] K. Zhu, "Modeling of the machining process," in *Smart Machining Systems: Modelling, Monitoring and Informatics*. Berlin, Germany: Springer, 2021, pp. 19–70.
[44] D. D'Addona et al., "Kohonen maps for chip form classification in turning," in *Proc. Intell. Prod. Mach. Syst.*, Elsevier, 2007, vol. 3, pp. 630–635.
[45] D. D'Addona et al., "Spectrum estimation and processing of cutting force sensor signals for chip form monitoring and classification," in *Proc. 4th Virtual Int. Conf. IPROMS*, 2008, pp. 555–560.
[46] *Tool-Life Testing With Single-Point Turning Tools*, Standard 3685:1993, ISO, Geneva, Switzerland, 1993.
[47] K. Jemielniak, R. Teti, J. Kossakowska, and T. Segreto, "Innovative signal processing for cutting force based chip form prediction," in *Proc. Intell. Prod. Mach. Syst.*, D. Pham, E. Eldukhri, and A. Soroka, Eds. Oxford, U.K.: Elsevier, 2006, pp. 7–12. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780080451572500092
[48] MATLAB, "Classification learner app," 2024. Accessed: Aug. 12, 2024. [Online]. Available: https://www.mathworks.com/help/stats/classification-learner-app.html
[49] STM32, "STM32Cube.AI (X-Cube-AI v9.0)," 2022. Accessed: Aug. 12, 2024. [Online]. Available: https://stm32ai.st.com/stm32-cube-ai

ALESSIA LAZZARO received the master's degree in electrical and electronics engineering from the University Mediterranea of Reggio Calabria (UNIRC), Reggio Calabria, Italy, in 2024. Her academic journey focused on the integration of TinyML and embedded systems, aiming to explore how machine learning can be efficiently implemented in hardware-limited environments.

Her research interests include the practical applications of TinyML in improving the performance and capabilities of embedded devices, and contributes to the evolving field by addressing the challenges of deploying AI in resource-constrained settings, striving for innovation in energy efficiency and processing power.

DORIANA MARILENA D'ADDONA received the Ph.D. degree in intelligent technology and systems for production automation with the University of Naples Federico II, Naples, Italy, in 2003.

She has authored or coauthored more than 140 publications, chaired several national and international conferences in the field of production engineering. She is a Associate Professor of manufacturing technology and systems with the Department of Chemical, Materials and Industrial Production Engineering, University of Naples Federico II. She received a Postdoc Fellow contract to work on intelligent computation for manufacturing technology and systems. Her main research interests include manufacturing processes and automation, advanced sensor applications for process monitoring, reconfigurable manufacturing machines and systems, intelligent computation, and machine learning for manufacturing technology and systems, and cognitive paradigms for Industry 4.0.

Dr. D'Addona is a Member of national and international scientific associations in the field of production engineering: CIRP, AITEM, IEEE, and the IEEE-IES Subcommittee on "Computer Vision and Human-Machine Interaction in Industrial and Factory Automation." She has taken part in a significant number of regional, national and international research projects and she is PI of two international projects.

MASSIMO MERENDA received the master's and Ph.D. degrees in electronic engineering from the University "Mediterranea" of Reggio Calabria, Reggio Calabria, Italy (UNIRC), in 2005 and 2009, respectively.

From 2003 to 2005, he was a Fellow with the Institute of Microelectronics and Microsystems, Italian National Research Council (IMM-CNR), Naples, Italy. Between 2011 and 2018, he was a Postdoctoral Researcher with UNIRC. From 2018 to 2021, he was a Researcher with the Department of Information Engineering, Infrastructure, and Sustainable Energy (DIIES), UNIRC, and the National Inter-University Consortium for Telecommunications (CNIT). From 2021 to 2022, he was a Senior Scientist with the Cooperative Digital Technologies Competence Unit, Austrian Institute of Technology (AIT), Vienna. Since October 2022, he has been a Senior Researcher with the DIIES Department of UNIRC. His research interests include the development of energy-efficient intelligent systems for the Internet of Things (IoT) and edge computing applications. This includes the design of CMOS integrated circuits, silicon sensors, and energy harvesting technologies. He specializes in creating embedded systems and RFID smart tags that facilitate advanced IoT solutions and contribute to the emerging field of the Internet of Conscious Things.

Open Access funding provided by 'Univ Mediterranea di Reggio Calabria' within the CRUI CARE Agreement