



From knowledge-based to big data analytic model: a novel IoT and machine learning based decision support system for predictive maintenance in Industry 4.0

Riccardo Rosati¹ · Luca Romeo^{1,2} · Gianalberto Cecchini³ · Flavio Tonetto³ · Paolo Viti⁴ · Adriano Mancini¹ · Emanuele Frontoni¹

Received: 10 September 2021 / Accepted: 30 April 2022 / Published online: 24 May 2022

© The Author(s) 2022, corrected publication 2022

Abstract

The Internet of Things (IoT), Big Data and Machine Learning (ML) may represent the foundations for implementing the concept of intelligent production, smart products, services, and predictive maintenance (PdM). The majority of the state-of-the-art ML approaches for PdM use different condition monitoring data (e.g. vibrations, currents, temperature, etc.) and run to failure data for predicting the Remaining Useful Lifetime of components. However, the annotation of the component wear is not always easily identifiable, thus leading to the open issue of obtaining quality labeled data and interpreting it. This paper aims to introduce and test a Decision Support System (DSS) for solving a PdM task by overcoming the above-mentioned challenge while focusing on a real industrial use case, which includes advanced processing and measuring machines. In particular, the proposed DSS is comprised of the following cornerstones: data collection, feature extraction, predictive model, cloud storage, and data analysis. Differently from the related literature, our novel approach is based on a feature extraction strategy and ML prediction model powered by specific topics collected on the lower and upper levels of the production system. Compared with respect to other state-of-the-art ML models, the experimental results demonstrated how our approach is the best trade-off between predictive performance (MAE: 0.089, MSE: 0.018, R^2 : 0.868), computation effort (average latency of 2.353 s for learning from 400 new samples), and interpretability for the prediction of processing quality. These peculiarities, together with the integration of our ML approach into the proposed cloud-based architecture, allow the optimization of the machining quality processes by directly supporting the maintainer/operator. These advantages may impact to the optimization of maintenance schedules and to get real-time warnings about operational risks by enabling manufacturers to reduce service costs by maximizing uptime and improving productivity.

Keywords IoT · Decision support system · Predictive maintenance · Machine learning · Random Forest

Introduction

The Internet of Things (IoT) and cyber-physical system (CPS) represent two fundamental pillars within the ubiq-

uitous scenario of Industry 4.0. These technologies lay the foundation for implementing the concept of intelligent production, smart products, and services. A realistic application of these technologies is predictive maintenance (PdM), which aims to address and diagnose in advance maintenance issues in order to minimize downtime and costs associated with it. Until now, factory managers and machine operators have carried out scheduled maintenance and regularly repaired machine parts to avoid downtime. In addition to consuming unnecessary resources and causing productivity losses, half of all preventive maintenance activities may be ineffective. Accordingly, the lack of maintenance will lead to more emergency breakdowns and downtime in production lines which affect the production capability by increasing operational cost, decreasing production rate, and reducing

✉ Riccardo Rosati
r.rosati@pm.univpm.it

¹ Department of Information Engineering, Marche Polytechnic University, Via Brecce Bianche 12, Ancona 60131, Italy

² Computational Statistics and Machine Learning, Istituto Italiano di Tecnologia, Via Enrico Melen, 83, 16152 Genoa, Italy

³ Sinergia Consulenze Srl, Viale Goffredo Mameli, 44, 61121 Pesaro, Italy

⁴ Benelli Armi Spa, Via della Stazione, 50, 61029 Urbino, Italy

the profit margin Burhanuddin et al. (2011). The downtime in production lines has more negative effects than the cost of repairing the failures. For example, if a filling machine fails in a production line, the end products will spill over Burhanuddin et al. (2011), thus leading to labour safety issues and business losses. It is therefore not surprising that PdM has quickly established itself as an industrial 4.0 use case and IoT-enabled PdM is attracting considerable investment and attention from industries and research Compare et al. (2019).

In this scenario, maintenance decision support systems (DSS) empowered by IoT, Big Data and Machine Learning (ML) assume a salient role to ensure maintainability and reliability of equipment in industries by transforming large datasets into knowledge and actionable intelligence Ayvaz and Alpay (2021); Chen et al. (2020); McArthur et al. (2018); Schmitt et al. (2020). Implementing industrial DSS to monitor the health of industrial processes, optimize maintenance schedules and get real-time warnings about operational risks enables manufacturers to reduce service costs, maximize uptime and improve productivity Schwendemann et al. (2021).

However, the application of PdM in production environments involves overcoming several unsolved challenges. These challenges include the need to aggregate several heterogeneous data gathered from sensors dealing with dynamic operating environments and the necessity of context-aware information targeted to a specific industrial domain Compare et al. (2019); Dalzochio et al. (2020). Therefore, one of the main challenges in this scenario is the difficulty to retrieve labeled training data of failure (i.e. annotated failure-related datasets). This kind of data is relevant for training supervised ML models to accurately and promptly identify problems and alarms.

Challenges and limitation of state-of-the-art

The basic idea behind PdM is to extrapolate/predict knowledge to determine the Remaining Useful Lifetime (RUL) of components. The majority of the state-of-the-art ML approaches for PdM (as reported in “[Related work](#)” section use different condition monitoring data (e.g. vibrations, currents, temperature, etc.) and run to failure data for predicting the RUL. However, the annotation of the component wear is not always easily identifiable and traced across different production cycles and operating conditions. Thus, the open issues include the difficulty of obtaining quality labeled data and interpreting it Vollert et al. (2021). A severe portion of available data has no annotations, presents missing values, and is poorly structured. This fact leads to the high demand to have available a huge amount of annotated failure-related datasets. The automatization, stability, and robustness of the annotation procedure is a challenge that still remains open

and represents a key issue for designing ML or Deep Learning (DL) approaches for accurately solving a PdM task.

Main contributions

The main contributions of this work to the field of novel applications and manufacturing support in Industry 4.0 are related to Big Data registration, processes and analyses. In particular, this paper aims to introduce a DSS for solving a PdM task by overcoming the above-mentioned challenges while focusing on a real industrial use case. The selected PdM task originated from a specific company’s demand¹ and the experimental setup (see Fig. 1) reflects a real industrial use case which includes advanced processing and measuring machines.

The proposed DSS comprised of five cornerstones by combining fundamental aspects of IoT, Big Data and ML: data collection, feature extraction, predictive model, cloud storage, and data analytics. Differently from the related literature, our novel approach is based on a feature extraction strategy and ML prediction model powered by specific topics² published in the MQTT broker and collected on the lower and upper levels of the production system. In particular, lower level topics are used to extract salient KPI predictors that are fed into a ML model while the upper-level topics reflect the ground-truth variables that are closely related to processing quality and thus to productivity losses and maintenance issues.

Comparing the proposed ML approach with respect to other state-of-the-art ML methods, the experimental results demonstrated how this strategy allows improving the predictive performance (in terms of MAE, MSE, and R^2) and at the same time the interpretability of the PdM task. Moreover, the lower computation effort allows to easily retrain the ML model also in the cloud environment.

The overall IoT-based sensing strategies and cloud storage architecture allow automatizing the annotation procedure by collecting a huge annotated dataset for training the ML model. On the other hand, the cloud storage framework lays the foundation for integrating an incremental learning strategy Rebuffi et al. (2017), where the ML model is continuously updated with new data, and it provides the data analytics for promptly monitoring the maintenance procedures.

The paper is organised as follows. In “[Related work](#)” section, we provide a review as well as the main differences with respect to state-of-the-art approaches in the field of PdM and

¹ Benelli Armi spa.

² Each client that publishes a message to the MQTT broker, includes a topic into the message. The topic is the routing information for the broker. Each client that wants to receive messages subscribes to a certain topic and the broker delivers all messages with the matching topic to the client. Therefore the clients don’t have to know each other, they only communicate over the topic.



(a) Machine center (MCM clock 5-axis machining center)



(b) Hexagon 3D coordinate measuring machine

Fig. 1 Experimental setup in the real industrial use case: advanced processing (a) and measuring (b) machines

ML. We describe the PdM task and the proposed real industrial use case in “[Real industrial use case PdM task](#)”. The description and implementation of the proposed DSS based on ML strategy is reported in “[Method](#)” section. In “[Experimental procedure](#)” section the evaluation procedure with respect to the state-of-the-art ML models is reported. The experimental results of our solution are reported in “[Experimental results](#)” section. Finally, the conclusions and future work are reported in “[Experimental results](#)” section.

Related work

The proposed approach is related to the general field of PdM. The state-of-the-art work in this field poses different challenges related to the downtime and maintenance-related costs and different solutions for improving production efficiency. According to the literature review proposed in Dalzochio et al. (2020), the state-of-the-art solutions to deal with this generic challenge can be divide into four groups: (i) *knowledge-based*, (ii) *Big Data* analytics, (iii) *Machine Learning* models and (iv) *Ontology and reasoning*. Different

from the knowledge-based approach, our strategy allows to completely automatize the feature extraction and prediction step for computing salient KPIs, which represent the input of our ML predictive model. Despite meta-heuristic optimization approaches (Abualigah, Yousri, et al., 2021; Abualigah, Diabat, Mirjalili, et al., 2021) are being introduced for solving a large scale combinatorial optimization problem related also to PdM tasks Shahbazi and Rahmati (2021); Shah et al. (2021), the proposal of IoT sensing technologies together with the designing of feature extraction stage and the deploying of ML model ensure to directly learn from data for solving the PdM task. This peculiarity allows monitoring, annotating, and consequently processing a large amount of heterogeneous data thus leading our approach relevant for both the ML and Big Data analytics area.

In the Big Data scenario, another open challenge is the difficulty to ensure accurate predictive performance and high interpretability at the same time Dalzochio et al. (2020). For example, the solution proposed in Hegedűs et al. (2018) provides a pre-processing step so that data are usable for PdM. Further work was done by Strauß et al. (2018), proposing a framework based on IoT and low-cost sensors that enables the monitoring and data acquisition on a heavy lift electric monorail system. On the contrary, our approach aims to acquire directly (i.e. without an additional pre-processing) a labeled and structured dataset that can be representative of the PdM task.

In the context of supervised learning, one important limitation is the lack of data showing the annotation of anomaly state behavior Adhikari et al. (2018); Dalzochio et al. (2020); Xu et al. (2019). In Gatica et al. (2016), the authors approached this problem by proposing a top-down strategy consisting of first understanding machine operation and then taking action to deal with the problem. On the contrary, here we build a data-driven model that can learn anomaly situations closely related to productivity losses. The model is able to generalize across different production cycles and operating conditions.

Starting from the fact that manufacturing plants are dynamic environments, both the lack Nuñez and Borsato (2018); Selcuk (2017) and the excess Gatica et al. (2016); Sarazin et al. (2019) of data heterogeneity may negatively affect the ML model. In this context, our approach tried to reduce the high dimension of collected data, by introducing a feature extraction stage based on hand-crafted features. Besides the fact that this step enhances the interpretability and explainability of the data, experimental results demonstrated how it increase the generalization power of the ML model.

Generally, state-of-the-art work test different ML models to evaluate which one is more suitable for a given situation. For example in Schmidt and Wang (2018), the authors presented a classification model to predict failure using as input

vibration limit value and used the accuracy of the models as an evaluation metric. Differently in Xu et al. (2019) and Nuñez and Borsato (2018), the computation cost for training the ML models is considered as an important aspect to take into account. In contrast with respect to the above-mentioned literature, our employed ML model represents the best trade-off between the accuracy prediction, computation effort and model interpretability for optimizing the machining quality processes in Industry 4.0 scenario.

In Peres et al. (2018) and Li et al. (2017), the artificial neural network model was employed for providing the fault prediction and identifying abnormal behaviors. Another work considers the implementation of auto-associative neural networks for finding irregularity in railways Liu et al. (2018). The recent advances of technology and the huge amount of data have laid the foundations to apply DL methodologies for solving PdM task Jin et al. (2017); Khan and Yairi (2018). The application of these solutions includes the implementation of a Recurrent Neural Network (RNN) and Long-Short-Term Memory (LSTM) Cachada et al. (2018); Rivas et al. (2019) for predicting failure by modeling spatio-temporal relationship across historical data. In the context of our proposed approach, the sequential DL approaches (such as RNN and LSTM) may represent an affordable predictive model by learning spatio-temporal features. However, the potential of DL approaches may be limited by the interpretability of the model Lipton (2018), which does not always allow to provide clues on how and why the algorithm achieved the selected prediction. Taking into account this consideration, as we shall see in the experimental section, our ML-based approach performs favorably over other standard (Multi layer perceptron) and sequential DL approaches (i.e. LSTM).

From ML perspective, most related to our work are the papers of Calabrese et al. (2020), Adhikari et al. (2018), Schmidt and Wang (2018), Zhou et al. (2018), Carbery et al. (2018), Ansari et al. (2020), which proposed the application of ML and DL model for predicting RUL and associated cost using sensor and event log data.

Common ML approaches for solving PdM task Bilski (2014); Calabrese et al. (2020); Schmidt and Wang (2018); Zhou et al. (2018) include the application of standard classifiers such as k-Nearest Neighbor (K-NN), Decision Tree (DT), Naive Bayes (NB), Support Vector Machine (SVM), Random Forest (RF), Boosting and XGBoost. In particular, the RUL prediction task was solved in Calabrese et al. (2020) using log-based data and XGboost algorithm. An evolution of ensemble learning (i.e. graph based ensemble learning) was presented in Zhou et al. (2018) for modeling the behaviour of different subsystems using different base learners. Graphical models based on Bayesian network and dynamics Bayesian network were also proposed in Carbery et al. (2018) and Ansari et al. (2020) for learning causal relationships among features and across time in terms of conditional probabili-

ties. In the latter, the graphical model is part of a framework designed to predict failures and to measure the impact of such a prediction on the quality of production planning processes and maintenance costs. Another class of ML models includes the application of auto-regressive models (e.g. Auto-regressive Moving Average) for predicting future behavior by using historical data. In Adhikari et al. (2018), the authors apply the auto-regressive integrated moving average model in a predictive maintenance framework to predict the remaining useful life of components.

The main differences with the proposed approach lie in: (i) the prediction of processing quality and anomaly situations that are quantitatively annotated in our training dataset by a 3D coordinate measuring machine; (ii) the integration of the proposed ML approach in our IoT platform that enables the collection of a huge amount of data and provide actionable decision recommendations for resolving productivity losses and maintenance issue; (iii) the combination of a feature extraction technique with a Random Forest (RF) regression model for ensuring the best trade-off between the accuracy prediction, computation effort, and model interpretability.

Real industrial use case PdM task

The proposed PdM task originated from a specific company's demand: the prediction of processing quality and anomaly situations during the machining of a tool. The input parameters were represented by topic at level 0, i.e. processing parameters (e.g. acceleration, speed, position) collected from two different machine centers (see Fig. 1a). The condition monitoring data were represented by topic at level 2 acquired by a robotic part loading system for coordinate measuring machine (see Fig. 1b). The overall flexible integrated manufacturing system includes two operator loading/unloading stations, two robot loading/unloading stations, one automated vertical parts store and one parts washing unit.

Machining centers

The processing parameters were acquired by two different machining centers (mc) (i.e. MCM³ clock 5-axis machining centers). On a 5-axis mc, the cutting tool moves across the X, Y and Z linear axes and rotates on the A and B axes to approach the workpiece from any direction. The mc can be configured for multitasking operations, such as milling, turning, grinding, boring, etc. Moreover, all mc can be configured with a single pallet, pallet exchanger, multi-pallet systems or integrated in a flexible manufacturing system (FMS). The level of automation can be changed or increased during the service life of the plant, providing considerable flexibility.

³ <https://www.mcmspa.it>.

Coordinate measuring machine for quality control task

The condition monitoring data were acquired by a robotic part loading system for coordinate measuring machine (CMM) (i.e. Hexagon⁴ Manufacturing Intelligence Robotic CMM part loading). The CMM system automatically identifies in real-time parts that are out of tolerance and triggers alarm situations. The CMM can be easily used by operators with minimal training as a cost-effective automated part loading system that increases the throughput of CMM and maximizes operational capacity.

Method

The proposed ML-based strategy is conceived to solve the above-mentioned PdM task by allowing the continuous collection of an annotated dataset and the provision of a data analytic interface for supporting the maintainer/operator. Figure 2 describes the overall architecture of the proposed DSS, which is comprised of five IoT and ML cornerstones:

- **Data collection:** the IoT sensing technology is based on the Message Queuing Telemetry Transport (MQTT) broker⁵ Andrew Banks Ed Briggs and Gupta (2019). The central concept in MQTT dispatcher is topics that collect processing parameters (e.g. acceleration, speed, position) of the machining centers at a lower level (level 0). Accordingly, the status and conditioning data collected by the Hexagon machine represents the topic at a higher level (level 2). All data are synchronized and are collected in a SQL database and then in Azure Blob cloud storage.
- **Feature extraction:** the Trapezoidal Numerical Integration (TNI) is performed to compute a Key Performance Indicator (KPI) for each processing parameter during each MCM production cycle.
- **Predictive model:** a RF regression model is applied in order to estimate the status and conditioning data using the collected processing parameters as predictors.
- **Cloud architecture:** the ML model is deployed as a docker container with an API endpoint for testing unseen acquired data. Azure ML service is used for providing a cloud-based environment for deploying and updating ML model. Azure Blob storage is adopted to store the ML model weights and the ML model outcomes (predicted error %).

⁴ <https://www.hexagonmi.com>.

⁵ MQTT is a lightweight telemetry protocol, coming from the world of M2M and now widely applied in IoT. The central communication point is the MQTT broker, which is in charge of dispatching all messages between the senders and the rightful receivers.

- **Data analytics:** the ML model outcomes are displayed in a GUI-based data analytic interface for supporting the maintainer/operator.

Data collection

The central sensing and communication point is the MQTT broker (Mosquito), which is in charge of dispatching all messages between the senders and the rightful receivers. The implementation of MQTT protocol in JFMX was performed by using the Paho Java library. Each client that published a message to the broker, includes a topic into the message, that represents the routing information for the broker. Each client that wants to receive messages subscribes to a certain topic and the broker delivers all messages with the matching topic to the client. Therefore the clients don't have to know each other, they only communicate over the topic. This architecture enables highly scalable solutions without dependencies between the data producers and the data consumers. The payload of messages are just a sequence of bytes, up to 256Mb, with no requirements placed on their format, and with MQTT protocol usually adding a fixed two bytes header to most messages. Other clients can subscribe to these messages and get updated by the broker when new messages arrive.

The central concept in MQTT to dispatch messages are topics. A topic is a simple string that can have more hierarchy levels. For example, a topic for sending status data of mc2 of an FMS is the following: JFMX/L1/fms/UNIT/mc1/STATUS. On one hand, the client can subscribe to the exact topic or on the other hand use a wildcard. The wildcard (+) allows arbitrary values for one hierarchy while the multilevel wildcard (#) allows to subscribe to more than one level (e.g. the entire subtree).

The MQTT topic organization allows to guarantee the Quality of Service (QoS). The MQTT protocol handles retransmission and ensures the delivery of the message, regardless how unreliable the underlying transport is. In addition, the client is able to choose the QoS level depending on its network reliability and application logic:

- **QoS 0—at most once** it guarantees a best effort delivery. A message won't be acknowledged by the receiver or stored and redelivered by the sender. This is often called "fire and forget" and provides the same guarantee as the underlying TCP protocol.
- **QoS 1—at least once** it is guaranteed that a message will be delivered at least once to the receiver. The sender will store the message until it gets an acknowledgement in form of a PUBACK command message from the receiver.
- **QoS 2—exactly once** it guarantees that each message is received only once by the counterpart. It is the safest and also the slowest QoS level. The guarantee is provided by two flows there and back between sender and receiver.

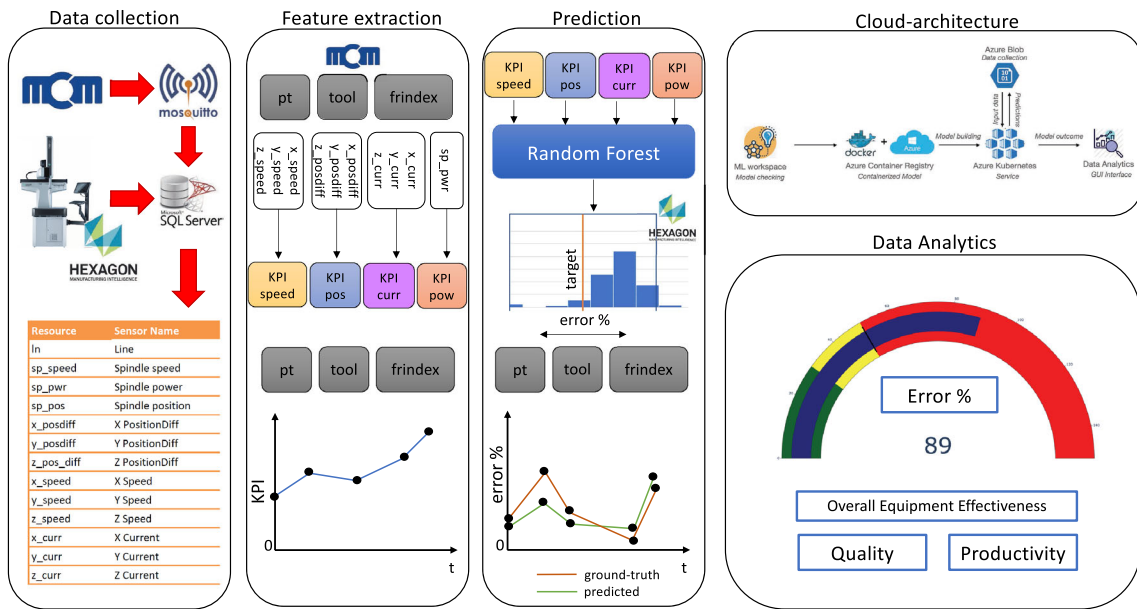


Fig. 2 Flowchart of the proposed approach: data collection, feature extraction, prediction phase based on Random Forest predictive model, cloud storage and data analytics

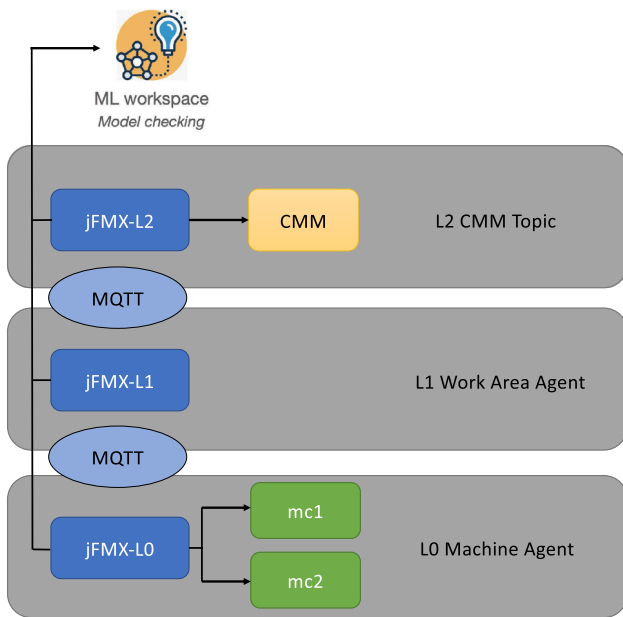


Fig. 3 Flowchart of the data layer: jFMX MQTT Namespace

jFMX MQTT Namespace The MQTT topic namespace was defined to manage interactions with the IoT application running on the jFMX gateway hierarchy. Figure 3 shows the hierarchy of the jFMX MQTT Namespace.

Based on this criterion, our IoT application running on an IoT gateway may be viewed in terms of the resources it owns and manages as well as the unsolicited events it reports:

Table 1 Topics related to the working step analyzer application embedded in the L0-Machine Agent

Field	Type	Description
ts	Date	Start date of the episode according to the clock of the mc
l2mc	String	mc code
l2wa	String	workArea code
ordNo	String	Order number
ptType	String	Identifier of the part type
opNo	String	Operation number
dType	String	Identifier of the workingstep message type ("wsEv": in case of sensor coming from mc, "accTrace": in case of sensor coming from Accelerometer)
Slot	Int	Subsection as indicated by the part program
pt	String	Identifier of the part machined
tl	String	Identifier of the tool formatted as <tooltype>/<tool serial number>
Life	Int	Life of the tool at the beginning of the step
Sensor	String	Name of the sensor
Unit	String	Measure unit for the specific sensor
Sampling	Int	Milliseconds sampling interval
dataObj	Object	Complex object containing two elements: n[int] segment of the acquisition, data: Array[Double] data acquired
mc	String	Machine name
wa	String	workArea name

- **account_name**: Identifies a group of devices and users. It can be seen as partition of the MQTT topic namespace. For example, access control lists can be defined so that users are only given access to the child topics of a given account_name.
- **client_id**: Identifies a single gateway device within an account (typically the MAC address of a gateway's primary network interface). The client_id maps to the Client Identifier (Client ID) as defined in the MQTT specifications.
- **app_id**: Unique string identifier for application (e.g., "L0" for mc topics, "L2" for CMM topics).
- **resource_id**: Identifies a resource(s) that is owned and managed by a particular application. Management of resources (e.g., sensors, actuators, local files, or configuration options) includes listing them, reading the latest value, or updating them to a new value. A resource_id is a hierarchical topic, where, for example, "fms/mc1/spindle/temp" may identify a temperature sensor and "fms/sh/Y/pos" a position sensor.

L0 Machine Center Topic The jFMX MQTT publisher is executed by the L0-Gateway (Flight Recorder) and delivers message related to different application included inside the L0-MachineAgent. All the topics published by the L0 MachineAgent is related to the processing parameters of mc and have an app_id defined as: JFMX/L0/workAreaName/unitName where the workAreaName is the absolute unique name of the workarea and the unitName is the name of the unit inside the workarea.

For our PdM task we refer to the topics related to the Working Step Analyzer application embedded in the L0-Machine Agent related to the two different mc (see Table 1)

The topics related to the dataObj field represent the considered processing parameters (see Table 2) acquired by mc (i.e. computer numerical control [CNC] and accelerometer). The CNC and accelerometer data were acquired by a sampling frequency of 24 and 100 Hz respectively.

L2 CMM Topic The L2 CMM topics represents the condition monitoring data that were acquired by a robotic part loading system for coordinate measuring machine (see Table 3). The condition monitoring data reflect the quality of processing in terms of the measured deviation (deviation) with respect to the optimal condition. The optimal condition highlighted no deviation compared to the planning processing. The alarm situation is triggered once the measured deviation overcomes the admitted tolerance.

All the L2 CMM Topic and L0 Machine Center Topic were synchronized by considering the physical tool (**tl**, identifier of the tool formatted as <tooltype>/<tool serial number>), the part machined (**pt**) and the type of processing (**frindex**). Although the system could consider all the type of processing we took into account the drilling procedure (i.e. FRINDEX = 10,20,30). This procedure has the intrinsic advantage of being standard, i.e. independent of the tl. The synchronized L2 CMM Topic together with the L0 Machine Center Topic were saved in a SQL database.

Feature extraction

All the computed KPIs (speed, pow, pos, curr) represent the predictors of the ML model for each observation/physical tool (i.e. specific triplet tl, pt and frindex). The output of the ML model was represented by the percentage measurement error (error %).

Table 2 Topics related to the dataObj field

Sensor (CNC)	Unit	Description
SP_SPEED	rpm	Spindle rotation speed
SP_POW	W	Spindle power consumption
X_AXIS_CURR	A	x-axis current consumption
Y_AXIS_CURR	A	y-axis current consumption
Z_AXIS_CURR	A	z-axis current consumption
B_AXIS_CURR	A	B-axis current consumption
A_AXIS_CURR	A	A-axis current consumption
X_AXIS_POS	μm	X-axis position
Y_AXIS_POS	μm	Y-axis position
Z_AXIS_POS	μm	Z-axis position
B_AXIS_POS	μm	B-axis position
A_AXIS_POS	μm	A-axis position
FRINDEX	None	Type of processing
Sensor (accelerometer)	Unit	Description
velMOD_RMS	mm/s	3 directions vibratory speed module

Table 3 L2 CMM topics

Field	Type	Description
ts	Date	Start date of the episode according to the clock of the mc
pt	String	Identifier of the part machined
tl	String	Identifier of the tool formatted as <tooltype>/<tool serial number>
FRINDEX	Int	type of processing
Measured	Double	Measured value
Deviation	Double	Measured deviation
Tolerance	Double	Admitted tolerance

Notation We let a candidate univariate time series of a specific sensor S collected from a CNC sensor as $X = \{x_0, x_1, \dots, x_T\}$ where T denotes the number of observations. Notice how this time series is relative to the signal of a specific sensor and relative to a specific triplet comprised of physical tool (tl), the part machined (pt) and the type of processing (frindex). We denote the *error%* as a direct quantitative measure about the machine quality and the *deviation* and *tolerance* the measured deviation and tolerance reported in Table 3.

Our feature extraction strategy is based on a geometric area analysis (GAA) and trapezoidal area estimation (TAE) procedure that is widely used for solving novelty and anomaly detection task Bulirsch and Stoer (2002); Moustafa et al. (2019). The relative KPI is computed by temporally normalizing the TAE as follows:

$$\begin{aligned}
 KPI &= \frac{1}{T} \int_1^T x_t dt \\
 &= \frac{1}{T} \sum_{t=1}^T \int_{t-1}^t x_t dt \\
 &\approx \frac{1}{2} \frac{1}{T} \sum_{t=1}^T (t - (t - 1)) [x_t - x_{t-1}] \tag{1}
 \end{aligned}$$

All the computed KPI for each specific sensor S is depicted in Table 4. For the position and current a global KPI was extracted by computing the euclidean distance of X, Y, Z axis.

For each observation, the *error %* was computed by considering the measured deviation and the associated tolerance

as follows:

$$\text{error\%} = \frac{\text{deviation}}{\text{tolerance}} \times 100 \tag{2}$$

In particular, an *error %* greater than 100 reflects an out of tolerance machining while an *error %* lower than 100 correspond to a machining that does not exceed the tolerance limits.

The final dataset consist of 438 observations/physical tools collected by two different mc from the 1st October 2019 to the 31st May 2020.

Predictive model

For solving the regression task we have taken into account predictive performance, interpretability, and predictive accuracy. These factors represent also the three fundamental requirements defined by the company for solving the PdM task. For this reason we selected the RF model for solving the regression task. RF represents a variant of bagging proposed by Breiman (2001) and consists of an ensemble of regression trees (RTs) (i.e., n° of RT) generated by independent identically distributed random vectors. RF is modeled by sampling from the observations, from the features (i.e., n° of features to be selected) and by changing two tree-parameters (i.e., $\max n^\circ$ of splits and $\max n^\circ$ of size) Kuncheva (2004). The idea behind this sampling is to maximize the diversity among trees, by sampling from the features set and from the data set as well. In particular, the random feature selection is carried out at each node of the tree, by choosing the best feature to split within a subset of the original feature set. In a standard classification problem, the idea is to built an ensemble of DT that aims to split the data into subsets that contain instances with similar values (homogenous). Since we aim to solve a regression problem, the best splitting features for each node was computed according to the sum of squared error.

Although RF allows learning a non-linear decision boundary, the RF originated as ensemble tree based model ensures an intuitive notion of interpretability: it allows providing a direct interpretation of the most discriminative KPIs. However, the degree of interpretability depends on the model size

Table 4 Extracted KPIs for specific sensor. KPIs are related to rotation speed (speed), power consumption (pow), position (pos) and current consumption (curr)

Sensor (CNC)	KPI
SP_SPEED	speed
SP_POW	pow
X,Y,Z_AXIS_POS	pos
X,Y,Z_AXIS_CURR	curr

(i.e., number of weak learners/regression tree and depth of the tree) Molnar et al. (2019). Hence, the interpretability of RF was encouraged by constraining the number of weak learners and the depth of the tree in the validation set. This lead also to control the computation effort for the training phase.

The importance of a specific KPI in the RF model to identify the percentage measurement error (error %) was measured according to a permutation of out-of-bag feature observation Boulesteix et al. (2012). A KPI is considered relevant to identify the error %, if permuting its values should affect the model error. On the other hand, if a KPI was not relevant, then permuting its values should not affect significantly the model error. The permutation importance of each feature is computed as: $1 - error$ (after permuting the feature values). Compared to the standard impurity-based importance the permutation approach is unbiased towards high cardinality features and measure directly the ability of feature to be useful to make prediction Janitza et al. (2018).

Cloud architecture

A container logic was adopted for packaging the ML application and all its dependencies, so the application runs reliably from one computing environment to another. A docker image is essentially a snapshot of a container. Microsoft Azure IoT portal was adopted for providing a cloud-based environment based on virtualized containers. This environment can ensure hardware and software isolation, flexibility, and inter-dependencies between the IoT devices and data collection, features extraction, and prediction phases. These properties are suitable for our industrial use case since the proposed DSS is currently designed to work with four operating machines and it provides the capability to be scaled up to collect a huge amount of data from different interconnected machines. This advantage also lies the foundations to continuously update the model, once a new machine is connected to the system.

Our architecture is depicted in Fig. 2 Cloud Architecture. We used a Python ML library for training and testing our feature extraction stage and RF model with respect to other state-of-the-art ML approaches. Afterward the feature extraction procedure and the containerized RF model was pushed to Azure Container Registry. During this step, we included the *azureml-monitoring* and *azureml-defaults* for enabling respectively the data collection feature and the deployment to Kubernetes. Consequently, the ML model was deployed to Azure Kubernetes Service (AKS). For our purpose we configured the AKS with 3 agent nodes of type Standard_D3_v2 (4 vCores), thus leading to a total of 12 vCores. Additionally, in our AKS configuration, we explicitly enable data collection (input data and predictions outcome). The L2 CMM Topic together with the L0 Machine Center Topic were exported from SQL database to azure blob storage, by allowing a continuous testing and update/retraining of the ML model once,

for instance, a drift situation was detected. All the prediction results together with the model weights (i.e. decision rule of the ensemble trees) were stored in the Azure blob storage.

Data analytics

A GUI interface was created to display the predicted error % over different tl, pt, frindex. In particular, the GUI was finalized to provide a timely indication to the machine operator when the error % exceeds a certain tolerance threshold that may be different for each tl, pt, frindex.

Additionally the Azure Application Insights instance was enabled for providing a high level overview of the deployed API in terms of featuring failed requests, response time, number of requests and availability. The log analytic feature of the Application insights allows to view and inspect the logs provided from our containerized model in terms of *stdout* and *stderr*.

Experimental procedure

Experimental comparisons

We decided to compare our RF based PdM approach with respect to other state-of-the-art ML approaches employed for solving PdM task. In particular we have considered the following models:

- Linear Regression with ridge penalty (LR ridge) Susto et al. (2012);
- Linear Regression with elastic net penalty (LR elastic) Susto et al. (2012);
- Regression Tree Romeo et al. (2020); Zhou et al. (2018);
- XGBoost Calabrese et al. (2020);
- SVM with Gaussian Kernel Bilski (2014)
- Multi layer perceptron (MLP) Jin et al. (2017); Khan and Yairi (2018)
- Long Short Term Memory (LSTM) Cachada et al. (2018); Rivas et al. (2019)

Experimental design

A 10-fold Cross Validation (10-CV) procedure was performed in order to evaluate the performance of the RF model. The hyperparameters were optimized by implementing a grid search in a nested 5-CV. Hence, each split of the outer CV loop was trained with the optimal hyperparameters (in terms of mean squared error) tuned in the inner CV loop. Despite this model checking procedure is expensive in terms of computation effort it allows to obtain an unbiased and robust performance evaluation Cawley and Talbot (2010).

Table 5 shows the different hyperparameters for the proposed ML models and all competitors' ML approaches, as well as the grid-search set.

For the LR ridge the λ penalty controls the 2-norm regularization. For the LR Elastic $\alpha = \lambda_1 + \lambda_2$ and $l1_ratio = \frac{\lambda_1}{\lambda_1 + \lambda_2}$ where λ_1 and λ_2 control separately the 1-norm and 2-norm regularizations.

- Mean Absolute Error (MAE): it measures the absolute difference between the predicted and the ground truth error %;
- Mean Squared Error (MSE): it measures the squared difference between the predicted and the ground truth error %;
- R^2 score (coefficient of determination): it is a proportion between the variability of the data and the correctness of the model used. It varies in range: $[-\infty; 1]$ Di Bucchinico (2008).

Evaluation metrics

The following metrics are considered to evaluate the predictive performance of the regression task:

The statistical significance of the R^2 score was evaluated at the 5% significance level with respect to the zero value. The R^2 score distribution over each CV fold was found to follow a normality distribution according to the Anderson-Darling

Table 5 Predictive performance and computation effort of ML approaches

Model	Hyp	Range
RF	n° of regression trees	{5, 10, 15, 20, 25}
	n° of features to select	{1, 2, 4}
LR ridge	λ	$\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$
LR elastic	α	$\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$
	l1_ratio	{ 10^{-4} , 10^{-3} , 10^{-2} , 0.1, 0.2, 0.3, 0.4, 0.5}
RT	Max depth	{5, 10, 15, 20, 25}
	Min n° of leaf size	{5, 10, 20, 50, 100}
XGBoost	Learning rate	{0.001, 0.01, 0.10}
	Max n° of estimators	{5, 10, 15, 20, 25}
	Max depth	{5, 10, 15, 20, 25, 50, 75}
	n° of features to select	{1, 2, 4}
SVM Gaussian	Box constraint	{1, 5, 10, 50, 100, 500, 10^3 , 5×10^3 , 5×10^4 , 10^4 }
	Kernel Scale	$\{10^{-2}, 0.1, 1, 10, 10^2, 10^3, 10^4\}$
MLP	Learning rate	$\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$
	n° of hidden layers	{1, 2, 4}
	n° of units	{4, 8, 16, 32}
LSTM	Learning rate	$\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$
	n° of hidden layers	{1, 2, 4}
	n° of units	{4, 8, 16}

Table 6 Predictive performance and computation effort of ML approaches

Model	MAE	MSE	R2	Training + validation (s)	Testing (s)
RF	0.089	0.018	0.868*	2.353 (0.328)	$< 10^{-3}$ (0)
LR ridge	0.166	0.054	0.591*	0.521 (0.007)	$< 10^{-3}$ (0)
LR elastic	0.166	0.054	0.591*	0.531 (0.058)	$< 10^{-3}$ (0)
RT	0.100	0.023	0.835*	0.310 (0.022)	$< 10^{-3}$ (0)
XGBoost	0.088	0.017	0.877*	80.740 (10.057)	$< 10^{-3}$ (0)
SVM Gaussian	0.151	0.047	0.648*	1.223 (0.101)	$< 10^{-3}$ (0)
MLP	0.161	0.051	0.618*	39.454 (0.651)	0.062 (0.002)
LSTM	0.156	0.044	0.667*	89.549 (2.618)	0.570 (0.023)

The best results for each metric are indicated in bold
 *Whether the R2 distribution over the 10-fold is significantly higher than 0

test ($A = 0.436$, $p = 0.246$). Hence, we used the parametric paired t-test ($\alpha = 0.05$) to compare the performance of the proposed approach with respect to state-of-the-art work.

Experimental results

In this section we show the experimental results for the the proposed DSS specifically tailored for solving PdM task. The prediction of the processing quality represents the main task we aim to solve. All the results related to the predictive performance and computation effort of the proposed approach with respect to the state-of-the-art approaches are depicted in “[Predictive performance](#)” and “[Computation effort](#)” sections, while the results related to the model interpretability, i.e. the most relevant KPIs, are shown in “[Interpretability](#)” section. More details on the implemented GUI for the proposed DSS are reported in “[Data analytics: GUI interfaces](#)” section.

Predictive performance

The predictive performance of the RF model is shown in Table 6. It can be noted that the best prediction results were obtained for our RF and XGboost model (R^2 score 0.868 and 0.877 respectively), while the LR model achieved the lowest predictive performance (R^2 score 0.591). Accordingly the RF and XGboost models show similar and competitive performance in terms of MAE (0.089 and 0.088 respectively) and MSE (0.018 and 0.017 respectively). R^2 score distribution of RF is significantly higher ($p < 0.05$) than ML based regression model (i.e. LR ridge, LR elastic net, RT, SVM Gaussian) and DL based regression model MLP and LSTM. In particular, the performance of sequential DL approaches might be limited by the low presence of a huge amount of annotated sequential data in this PdM scenario, in order to learn spatio-temporal dependencies.

Figure 4 shows the comparison between the predicted error % from RF and its real values obtained from the L2 CMM Topic. We focused on a subset of 42 testing samples (one fold of CV-10 procedure).

Computation effort

Taking into account the high performance achieved by the proposed approach, we decided to test the computation effort with respect to other state-of-the-art ML approaches. Table 6 compares the time effort (test and training+validation stage) of the proposed RF algorithm with respect to other state-of-the-art algorithms. All the experimental comparisons were performed on Intel Core i7-4790 CPU 3.60 GHz with 16 GB

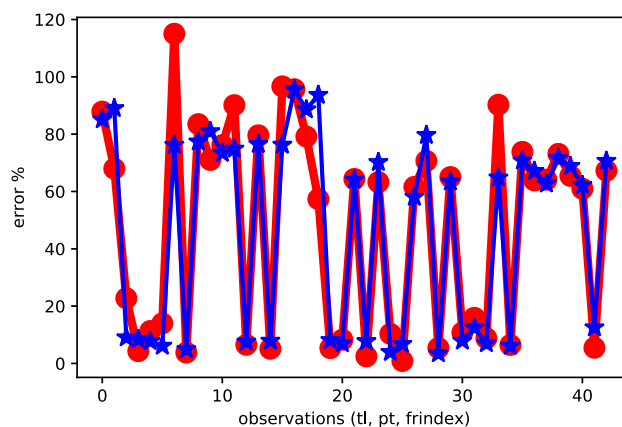


Fig. 4 Predictive performance of RF for the estimation of the error % over a subset of observations (tl, pt, frindex): red line ground truth, blue line prediction (Color figure online)

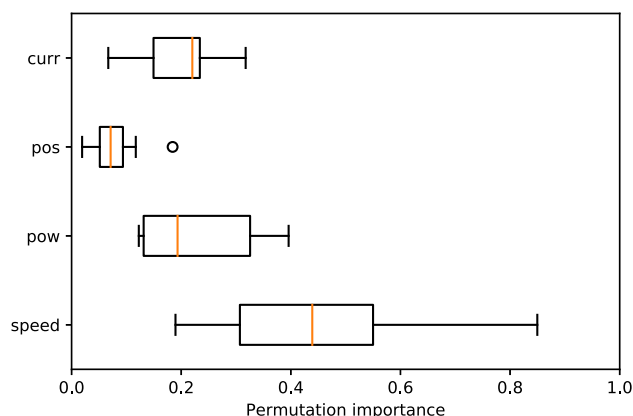


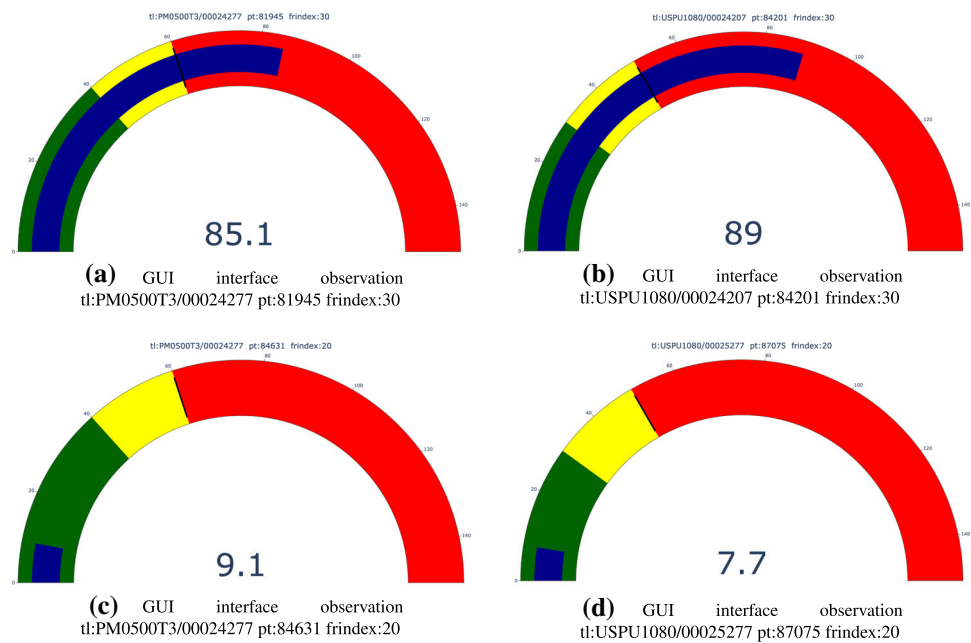
Fig. 5 Feature/permutation importance of RF model (Color figure online)

of RAM and NVIDIA GeForce GTX 970. Although the predictive performance of RF is similar to XGboost, the training and validation of RF model are significantly ($p < 0.05$) faster than XGboost with a gain of 34x. This peculiarity ensures the possibility to retrain the ML model in the cloud with an average latency of 2.353 s for learning from around 400 new samples. At the same time the RF prediction latency may be neglected $< 10^{-3}$ s and the RF model can give a timely and consistent prediction.

Interpretability

The interpretability of the proposed RF model was measured according to the feature/permutation importance (Figure 5). The speed KPI achieved, on average, the highest permutation importance score, thus highlighting the most discriminative power of this KPI with respect to the other features. The feature importance together with the predicted error % values are the salient ML outcomes of the proposed DSS for sup-

Fig. 6 Examples of GUI interface of the proposed DSS for four different observations (tl, pt, frindex): the predicted error % is represented by blue line and reported below the gauge chart, the black line represents the admissible tolerance threshold that can change across different triplets. Notice how an error % greater than the admissible threshold (red bar) represents a significant machining error (i.e. alarm event), while an error % that falls within the yellow bar correspond to a potential risk situation (i.e. machining error below but close to tolerance limits). The green bar reflect how the machining is properly executed within the tolerance limit (Color figure online)



porting the maintainer/operator during the machining quality task. For instance, taking into account the predicted error %, the operator may exploit preventive action in order to avoid future errors during the machine processing. At the same time, the localization of the most discriminative KPI may address the human operator to detect the source of the error, while optimizing the overall equipment effectiveness, productivity, and quality of production.

Data analytics: GUI interfaces

Figure 6 shows an examples of GUI interfaces of the proposed DSS for four different observations (tl, pt, frindex). We represent the predicted error % and the tolerance limits, which can be different for each observation. In particular, a predicted error % greater than the admissible threshold (red bar) represents a significant machining error (i.e. alarm event), while an error % that falls within the yellow bar correspond to a potential risk situation (i.e. machining error below but close to tolerance limits). The green bar reflect how the machining is properly executed within the tolerance limit.

Taking into account the high predictive results and the interpretability of the proposed approach, our GUI interface is not limited to show only the predicted error % (i.e. Fig. 6). In fact, we go further by supporting the operator by showing the average value of the KPI predictors (see Fig. 7) for a specific tl across different pt and frindex. Additionally, the trend of KPI predictors together with the life parameter (i.e. life of the tool at the beginning of the step) is displayed for a specific tl and frindex across pt and across time in a separated dashboard, as shown in Fig. 8. These GUI interfaces may empower the overall machining quality process by support-

ing the operators to (i) predict alarm situation (i.e. significant machining error) and (ii) interpret and localize the source of the error by focusing on the average and temporal value of the most discriminative KPI predictors (see Fig. 5).

Conclusions

We have shown that our DSS approach is effective in solving the machining quality and PdM task in a real industrial use case. The main pillars of the proposed DSS consist of data collection, feature extraction, predictive model, cloud storage, and data analytics. The strict collaboration between the company and University allowed to design, validate and test the proposed approach in a real industrial PdM use case task, by considering the dataset described in “[Feature extraction](#)” section. Taking into account the achieved experimental results, we demonstrated the effectiveness of our theoretical frameworks into a real industrial environment. In fact, our DSS approach was demonstrated to be the best trade-off between predictive performance, computation effort, and interpretability. These peculiarities together with the integration of our ML approach into the proposed cloud-based architecture would allow the optimization of the machining quality processes by directly supporting the maintainer/operator. These advantages may impact to the optimization of maintenance schedules and to get real-time warnings about operational risks by enabling manufacturers to reduce service costs by maximizing uptime and improving productivity.

A current limitation is represented by the employed ML model, which is re-trained every time from scratch when a

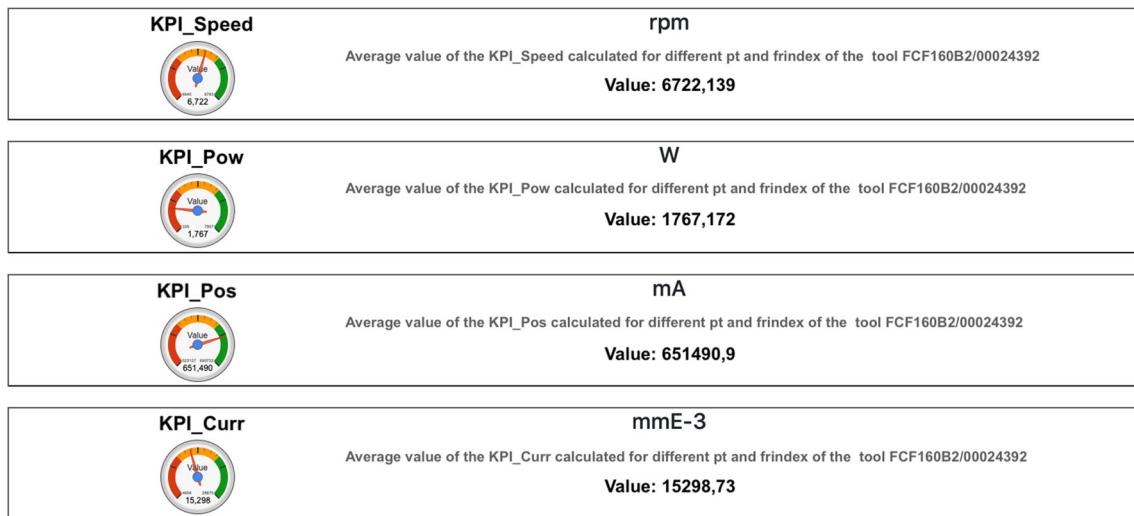


Fig. 7 Example of the proposed GUI interface displaying the average value of the KPI predictors for a specific tl across different pt and frindex

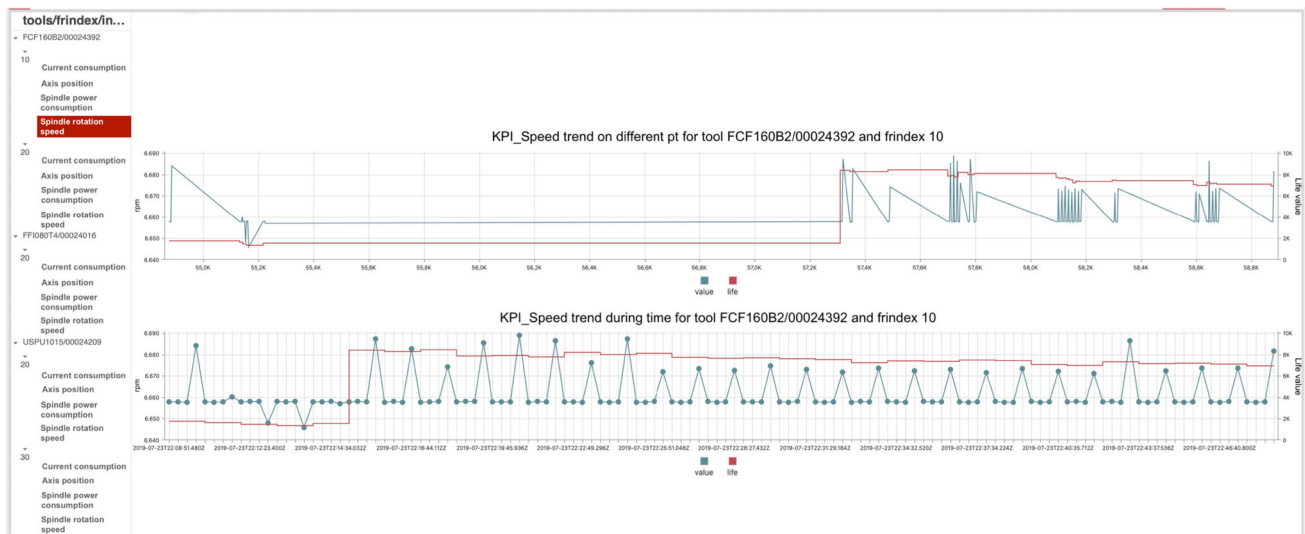


Fig. 8 Example of the proposed GUI interface displaying the trend of KPI predictors (blue line) together with the life parameter (red line) (i.e. life of the tool at the beginning of the step) for a specific tl and frindex across pt (top graph) and across time (bottom graph)

certain amount of new data is stored. As future work, we aim to integrate a fully-automated incremental learning procedure by updating continuously the model parameter Chai and Zhao (2020). The new data continuously acquired and stored over time in the proposed cloud framework can be used to refine ML model and improve its predictive performance according to an incremental learning procedure. As a further limitation, the proposed model only works for certain tools installed on the processing machines. However, as future work, for the proposed ML model we aim to improve the generalization performance of the proposed approach across different tools and types of processing. In addition, meta-heuristic algorithms (Abualigah et al., 2022;

Abualigah, Diabat, Sumari, et al., 2021) could be implemented to help in selecting the optimal hyperparameters for ML model to improve the stability and the testing predictive performance Shah et al. (2021); Zhang et al. (2020). For the full applicability of PdM in all the company’s tasks, another future work direction could be addressed to build integrated cost-benefit models that include the impact and the benefit of our approach on the entire asset management of the company Compare et al. (2019).

Acknowledgements The authors would like to thank Benelli Armi Spa and Sinergia Consulenze Srl for their contribution.

Funding Open access funding provided by Università Politecnica delle Marche within the CRUI-CARE Agreement. This work was supported within the research agreement between Università Politecnica delle Marche and Benelli Armi Spa for the "4USER Project" (User and Product Development: from Virtual Experience to Model Regeneration) funded on the POR MARCHE FESR 2014-2020-ASSE 1-OS 1-ACTION 1.1-INT. 1.1.1. Promotion of industrial research and experimental development in the areas of smart specialisation -LINEA 2 -Bando 2019, approved with DDPF 293 of 22/11/2019.

Declarations

Conflict of interest The authors declare that they do not have any conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Abualigah, L., Abd Elaziz, M., Sumari, P., Geem, Z. W., & Gandomi, A. H. (2022). Reptile search algorithm (RSA): A nature-inspired meta-heuristic optimizer. *Expert Systems with Applications*, *191*, 116158.
- Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., & Gandomi, A. H. (2021). The arithmetic optimization algorithm. *Computer Methods in Applied Mechanics and Engineering*, *376*, 113609.
- Abualigah, L., Diabat, A., Sumari, P., & Gandomi, A. H. (2021). Applications, deployments, and integration of internet of drones (IOD): a review. *IEEE Sensors Journal*, *21*(22), 25532–25546.
- Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A. A., Al-qaness, M. A., & Gandomi, A. H. (2021). Aquila optimizer: A novel meta-heuristic optimization algorithm. *Computers & Industrial Engineering*, *157*, 107250.
- Adhikari, P., Rao, H. G. & Buderath, M. (2018). Machine learning based data driven diagnostics & prognostics framework for aircraft predictive maintenance. In *10th international symposium on NDT in aerospace*.
- Andrew Banks Ed Briggs, K. B., & Gupta, R. (2019). Mqtt version 5.0 oasis standard. Retrieved March 7, 2019, from <https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html>
- Ansari, F., Glawar, R. & Sihni, W. (2020). Prescriptive maintenance of CPPS by integrating multimodal data with dynamic Bayesian networks. In *Machine learning for cyber physical systems* (pp. 1–8). Springer.
- Ayvaz, S., & Alpay, K. (2021). Predictive maintenance system for production lines in manufacturing: A machine learning approach using IoT data in real-time. *Expert Systems with Applications*, *173*, 114598.
- Bilski, P. (2014). Application of support vector machines to the induction motor parameters identification. *Measurement*, *51*, 377–386.
- Boulesteix, A.-L., Janitza, S., Kruppa, J., & König, I. R. (2012). Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. *Data Mining and Knowledge Discovery*, *2*(6), 493–507.
- Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.
- Bulirsch, R. & Stoer, J. (2002). *Introduction to numerical analysis* (Vol. 3). Springer.
- Burhanuddin, M. A., Halawani, S. M. & Ahmad, A. (2011). An efficient failure-based maintenance decision support system for small and medium industries. In *Efficient decision support systems: practice and challenges from current to future 195*. InTechOpen.
- Cachada, A., Barbosa, J., Leitão, P., Grcaldcs, C. A., Deusdado, L., Costa, J., Teixeira, C., Teixeira, J., Moreira, A. H. J., Miguel, P., & Romero, L. (2018). Maintenance 4.0: Intelligent and predictive maintenance system architecture. In *2018 IEEE 23rd international conference on emerging technologies and factory automation (ETFA)* (Vol. 1, pp. 139–146).
- Calabrese, M., Cimmino, M., Fiume, F., Manfrin, M., Romeo, L., Ceccacci, S., & Kapetis, D. (2020). Sophia: An event-based IoT and machine learning architecture for predictive maintenance in industry 4.0. *Information*, *11*(4), 202.
- Carbery, C. M., Woods, R. & Marshall, A. H. (2018). A Bayesian network based learning system for modelling faults in large-scale manufacturing. In *2018 IEEE international conference on industrial technology (ICIT)* (pp. 1357–1362).
- Cawley, G. C., & Talbot, N. L. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, *11*(7), 2079–2107.
- Chai, Z., & Zhao, C. (2020). Multiclass oblique random forests with dual-incremental learning capacity. *IEEE Transactions on Neural Networks and Learning Systems*, *31*(12), 5192–5203.
- Chen, C., Liu, Y., Wang, S., Sun, X., Di Cairano-Gilfedder, C., Titmus, S., & Syntetos, A. A. (2020). Predictive maintenance using cox proportional hazard deep learning. *Advanced Engineering Informatics*, *44*, 101054.
- Compare, M., Baraldi, P., & Zio, E. (2019). Challenges to IoT-enabled predictive maintenance for industry 4.0. *IEEE Internet of Things Journal*, *7*(5), 4585–4597.
- Dalzochio, J., Kunst, R., Pignaton, E., Binotto, A., Sanyal, S., Favilla, J., & Barbosa, J. (2020). Machine learning and reasoning for predictive maintenance in industry 4.0: Current status and challenges. *Computers in Industry*, *123*, 103298.
- Di Buccianico, A. (2008). Coefficient of determination (R^2). In *Encyclopedia of statistics in quality and reliability 1*. Wiley.
- Gatica, C. P., Koester, M., Gaukster, T., Berlin, E. & Meyer, M. (2016). An industrial analytics approach to predictive maintenance for machinery applications. In *2016 IEEE 21st international conference on emerging technologies and factory automation (ETFA)* (pp. 1–4).
- Hegedűs, C., Ciancarini, P., Frankó, A., Kancilija, A., Moldován, I., Papa, G., Poklukar, S., Riccardi, M., Sillitti, A., & Varga, P. (2018). Proactive maintenance of railway switches. In *2018 5th international conference on control, decision and information technologies (CoDIT)* (pp. 725–730).
- Janitza, S., Celik, E., & Boulesteix, A.-L. (2018). A computationally fast variable importance test for random forests for high-dimensional data. *Advances in Data Analysis and Classification*, *12*(4), 885–915.
- Jin, L., Wang, F. & Yang, Q. (2017). Performance analysis and optimization of permanent magnet synchronous motor based on deep learning. In *20th international conference on electrical machines and systems (ICEMS)* (pp. 1–5).
- Khan, S., & Yairi, T. (2018). A review on the application of deep learning in system health management. *Mechanical Systems and Signal Processing*, *107*, 241–265.

- Kuncheva, L. I. (2004). *Combining pattern classifiers: Methods and algorithms*. Wiley.
- Li, Z., Wang, Y., & Wang, K.-S. (2017). Intelligent predictive maintenance for fault diagnosis and prognosis in machine centers: Industry 4.0 scenario. *Advances in Manufacturing*, 5(4), 377–387.
- Lipton, Z. C. (2018). The myths of model interpretability. *Queue*, 16(3), 31–57.
- Liu, Z., Jin, C., Jin, W., Lee, J., Zhang, Z., Peng, C. & Xu, G. (2018). Industrial AI enabled prognostics for high-speed railway systems. In *2018 IEEE international conference on prognostics and health management (ICPHM)* (pp. 1–8).
- McArthur, J., Shahbazi, N., Fok, R., Raghobar, C., Bortoluzzi, B., & An, A. (2018). Machine learning and BIM visualization for maintenance issue classification and enhanced data collection. *Advanced Engineering Informatics*, 38, 101–112.
- Molnar, C., Casalicchio, G. & Bischl, B. (2019). Quantifying interpretability of arbitrary machine learning models through functional decomposition. *Ulmer Informatik-Berichte* 41. Preprint.
- Moustafa, N., Slay, J., & Creech, G. (2019). Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks. *IEEE Transactions on Big Data*, 5(4), 481–494.
- Núñez, D. L., & Borsato, M. (2018). Ontoprog: An ontology-based model for implementing prognostics health management in mechanical machines. *Advanced Engineering Informatics*, 38, 746–759.
- Peres, R. S., Rocha, A. D., Leitao, P., & Barata, J. (2018). Idarts-towards intelligent data analysis and real-time supervision for industry 4.0. *Computers in Industry*, 101, 138–146.
- Rebuffi, S.-A., Kolesnikov, A., Sperl, G. & Lampert, C. H. (2017). ICARL: Incremental classifier and representation learning. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2001–2010).
- Rivas, A., Fraile, J. M., Chamoso, P., González-Briones, A., Sittón, I. & Corchado, J. M. (2019). A predictive maintenance model using recurrent neural networks. In *International workshop on soft computing models in industrial and environmental applications* (pp. 261–270).
- Romeo, L., Loncarski, J., Paolanti, M., Bocchini, G., Mancini, A., & Frontoni, E. (2020). Machine learning-based design support system for the prediction of heterogeneous machine parameters in industry 4.0. *Expert Systems with Applications*, 140, 112869.
- Sarazin, A., Trupitil, S., Montarnal, A. & Lamothe, J. (2019). Toward information system architecture to support predictive maintenance approach. In *Enterprise interoperability viii* (pp. 297–306). Springer.
- Schmidt, B., & Wang, L. (2018). Predictive maintenance of machine tool linear axes: A case from manufacturing industry. *Procedia Manufacturing*, 17, 118–125.
- Schmitt, J., Böning, J., Borggräfe, T., Beiting, G., & Deuse, J. (2020). Predictive model-based quality inspection using machine learning and edge cloud computing. *Advanced Engineering Informatics*, 45, 101101.
- Schwendemann, S., Amjad, Z., & Sikora, A. (2021). A survey of machine-learning techniques for condition monitoring and predictive maintenance of bearings in grinding machines. *Computers in Industry*, 125, 103380.
- Selcuk, S. (2017). Predictive maintenance, its implementation and latest trends. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 231(9), 1670–1679.
- Shah, P., Sekhar, R., Kulkarni, A. J. & Siarry, P. 2021. *Metaheuristic algorithms in industry 4.0*. CRC Press.
- Shahbazi, B., & Rahmati, S. H. A. (2021). Developing a flexible manufacturing control system considering mixed uncertain predictive maintenance model: A simulation-based optimization approach. *Operations Research Forum*, 2, 1–43.
- Strauß, P., Schmitz, M., Wöstmann, R. & Deuse, J. (2018). Enabling of predictive maintenance in the brownfield through low-cost sensors, an IIOT-architecture and machine learning. In *2018 IEEE international conference on big data (Big Data)* (pp. 1474–1483).
- Susto, G. A., Schirru, A., Pampuri, S. & Beghi, A. (2012). A predictive maintenance system based on regularization methods for ion-implantation. In *2012 semi advanced semiconductor manufacturing conference* (pp. 175–180).
- Vollert, S., Atzmueller, M. & Theissler, A. (2021). Interpretable machine learning: A brief survey from the predictive maintenance perspective. In *2021 26th IEEE international conference on emerging technologies and factory automation (ETFA)* (pp. 01–08).
- Xu, Y., Sun, Y., Liu, X., & Zheng, Y. (2019). A digital-twin-assisted fault diagnosis using deep transfer learning. *IEEE Access*, 7, 19990–19999.
- Zhang, P., Wu, H.-N., Chen, R.-P., & Chan, T. H. (2020). Hybrid meta-heuristic and machine learning algorithms for tunneling-induced settlement prediction: A comparative study. *Tunnelling and Underground Space Technology*, 99, 103383.
- Zhou, C. & Tham, C.-K. (2018). Graphel: A graph-based ensemble learning method for distributed diagnostics and prognostics in the industrial internet of things. In *2018 IEEE 24th international conference on parallel and distributed systems (ICPADS)* (pp. 903–909).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.