processes

Article

A Tool Condition Monitoring System Based on Low-Cost Sensors and an IoT Platform for Rapid Deployment

Johanna Marie Failing, José V. Abellán-Nebot *, Sergio Benavent Nácher, Pedro Rosado Castellano and Fernando Romero Subirón

Department of Industrial Systems Engineering and Design, Universitat Jaume I, Av. Vicent Sos Baynat, 12071 Castellón de la Plana, Spain

* Correspondence: abellan@uji.es; Tel.: +34-964-728186

Abstract: Tool condition monitoring (TCM) systems are key technologies for ensuring machining efficiency. Despite the large number of TCM solutions, these systems have not been implemented in industry, especially in small- and medium-sized enterprises (SMEs), mainly because of the need for invasive sensors, time-consuming deployment solutions and a lack of straightforward, scalable solutions from the laboratory. The implementation of TCM solutions for the new era of the Industry 4.0 is encouraging practitioners to look for systems based on IoT (Internet of Things) platforms with plug and play capabilities, minimum interruption time during setup and minimal experimental tests. In this paper, we propose a TCM system based on low-cost and non-invasive sensors that are plug and play devices, an IoT platform for fast deployment and a mobile app for receiving operator feedback. The system is based on a sensing node by Arduino Uno Wi-Fi that acts as an edge-computing node to extract a similarity index for tool wear classification; a machine learning node based on a BeagleBone Black board that builds the machine learning model using a Python script; and an IoT platform to provide the communication infrastructure and register all data for future analytics. Experimental results on a CNC lathe show that a logistic regression model applied on the machine learning node can provide a low-cost and straightforward solution with an accuracy of 88% in tool wear classification. The complete solution has a cost of EUR 170 and only a few hours are required for deployment. Practitioners in SMEs can find the proposed approach interesting since fast results can be obtained and more complex analysis could be easily incorporated while production continues using the operator's feedback from the mobile app.

Keywords: tool condition monitoring; industrial internet of things; logistic regression; IoT platform; modeling

check for updates

Citation: Failing, J.M.; Abellán-Nebot, J.V.; Benavent Nácher, S.; Rosado Castellano, P.; Romero Subirón, F. A Tool Condition Monitoring System Based on Low-Cost Sensors and an IoT Platform for Rapid Deployment. *Processes* **2023**, *11*, 668. https://doi.org/10.3390/pr11030668

Academic Editor: Uros Zuperl

Received: 19 November 2022
Revised: 10 February 2023
Accepted: 20 February 2023
Published: 22 February 2023

1. Introduction

Machining processes are widely used manufacturing processes where excess material from the surface of a workpiece is removed using different cutting tools. Turning and milling operations, the most common group of operations in conventional machining processes, are located in almost any manufacturing shop floor and their performance has a clear impact on part quality and manufacturing cost.

According to the literature, cutting tool failures may represent about 20% of the downtime of a machining system, and it is estimated that the expense of cutting tools and their maintenance amounts to about 3–12% of overall manufacturing cost [1,2]. Common industrial practices apply early tool replacement strategies to avoid these issues, which leads to cutting tool life reduction, increased machining cost and higher environmental impacts. In fact, in [3], it is stated that only 57% of tools are used up to their full tool life ability, although other researchers reported more conservative values around 80% [4].

Although there exist some empirical/mechanistic models to predict tool life, since tool wear is considered stochastic, it is generally difficult to predict [5]. In fact, cutting tool life may vary greatly among cutting tools, even if they are the same type produced by the same manufacturer. For instance, within a single batch, the tool life of Russian production may vary widely by 15–40% [6].

Government initiatives for sustainability demand more efficient practices in order to minimize energy consumption and maximize cutting tool life. In the field of increasing the efficiency of use for cutting tools, several research directions can be found, such as new coating materials and use of texturized tools [7], new cooling fluids such as cryogenic coolants [8], the adoption of advanced machining strategies such as high-speed machining [9] and the implementation of tool condition monitoring (TCM) systems to monitor cutting tool state to optimize cutting parameters and tool change policies [10].

The TCM systems have been widely investigated in the past decades and can significantly reduce downtime by effective predictive maintenance actions, extending cutting tool life and improving productivity, quality and sustainability. These systems are based on different real-time sensors such as dynamometers, accelerometers, acoustic emission (AE) sensors, current/power sensors, etc., and they can estimate tool wear progression, classify the state of the tool as worn/not worn and reveal chatter or tool breakage occurrence. The estimation is commonly realized by a predictive model based on regression or AI models. Several reviews can be found about TCM in machining, where detailed information is given about the sensor systems, signal processing, feature extraction and modeling techniques [10–12].

As reported in the reviews, most of the research conducted on TCM in machining is based on expensive and accurate sensors with advanced modeling techniques, especially with artificial intelligent (AI) techniques and machine learning approaches such as Artificial Neural Network (ANN), Support Vector Machine (SVM), Bayesian Network (BN), Neuro-Fuzzy (ANFIS) models and, in the past 5 years, deep learning techniques [11,13]. Most of the case studies and experimental validation found in the literature are related to laboratory tests, showing high accuracy and reliability. However, despite the potential benefits of TCM in industry (e.g., in [14], it is stated that TCM could generate 75% downtime reduction and 30% maintenance cost saving) and the industrial lead research projects from past years [15], the implementation of these systems in industry is limited, especially for small- and medium-sized enterprises (SMEs). Some of the current issues that prevent the implementation of TCM in industry are:

- High cost of the equipment and incompatible configuration.
- Time-consuming deployment due to specific design of experiments and model training and validation activities which interrupt normal production.
- Use of invasive sensors.
- Hardware and software solutions at laboratory scale that are not straightforward solutions for industry, such as computers on shop floors with a data acquisition system or excessively complex models that are difficult for practitioners to use.

Fortunately, with the advent of the Industry 4.0 practices and funds from national governments for digitalization, companies are more likely to implement sensor networks to acquire real-time information to increase manufacturing efficiency. The deployment of the Industrial Internet of Things (IIoT) and cloud computing is becoming a reality, and the implementation of TCM systems based on low-cost and plug and play sensors with IoT platforms and mobile apps can be conducted in a faster, less expensive and less invasive manner than before.

Recent research works have dealt with these new opportunities for TCM systems. In [16], the authors proposed an IoT-based architecture for monitoring the milling process and inspecting the resulting part quality. NodeMCU devices, wireless modules similar to Arduino boards, were used to connect the sensing units and this information was later transferred to the cloud through a Raspberry Pi. The monitoring control interface was established with the interpreted STEP-NC information which enables any change in cutting parameters.

Wu et al. [17] proposed a cloud-based parallel machine learning system to deal with monitoring systems where large volumes of training data are required to make accurate predictions. The proposed system showed a MapReduce-based parallel random forests (PRFs) algorithm to predict tool wear in milling operations. Recently, Zuperl et al. [18] presented a cloud-based system for online monitoring of tool conditions. The monitoring system introduces an optical system to extract data about the chip size and sends them to the IoT application, where they are used as indicators for the determination of tool conditions. At the cloud level, an ANN model is trained to identify cutting force trends and protect the tool against excessive loading by correcting process parameters. The paper reported a classification accuracy of 85.3% with no error in the identification of tool breakage.

A TCM was proposed in [19], where a prototype for prognosis in a CNC machine tool and industrial pumps was presented. The prototype was built based on the Predix Machine, an Industrial Internet software, and different sensor nodes were deployed for vibration, acoustic emission, force and current measurement. Each sensor consisted of a microprocessor with an analog-to-digital converter and a ZigBee wireless module. A machine learning algorithm (random forests) was implemented on a public cloud, although all the experimental data for training and validation were obtained offline.

To avoid latency problems with cloud computing, edge-computing approaches have also been presented [20–22]. In [20], a TCM was implemented for repetitive milling processes based on an edge-computing algorithm to avoid the latency issue of cloud computing. At the edge level, the whole profile from a signal acquired by a sensor device was compared through a similarity analysis with respect to a calibration profile. According to the value of the similarity parameter, the system shows how close the cutting tool is to the worn state. The system was proved to be accurate enough under this type of milling scenario. A similar approach is presented in [21], where instead of a similarity analysis, a correlation coefficient was calculated from a regression analysis between two different cycles of time series from power consumption data. The correlation value was found to change notably when severe tool wear arises.

Connectivity of machine tools to the IoT platform has also been an issue for TCM systems. Monitoring the current cutting parameters and changing them according to sensor information can be barriers to the deployment of TCM systems in some CNC machine tools. MTConnect [23] and OPC UA [24] are recent real-time data exchange solutions for manufacturing systems. MTConnect is implemented as a web service and it is specifically designed for CNC machine tools with some predefined data structures and rules, and it is easily accessible to any device that is connected to the machine network. OPC UA is a more generic information modeling method to cover the communication among a wide range of industrial equipment. Both MTConnect and OPC UA are open and royalty-free communication standards designed for industrial automation and allow users to obtain process-related data in real time such as spindle speed, angular acceleration and velocity, load and so on. In [25], a TCM system is proposed using as sensor measurements the spindle motor load, *X*-axis motor load and *Z*-axis motor load acquired from the machine tool using an MTConnect adapter. Therefore, instead of expensive sensors and invasive sensors, the proposed system uses signals already available to the machine through a standard Internet protocol. The experiment was conducted with new and worn tools, and the data sets were used for conducting different machine learning approaches such as optimal linear associative memory (OLAM) neural networks and self-organizing deep learning methods. The results showed high accuracy to detect worn tools using self-organized networks, with an accuracy of 93% versus 65% for the OLAM networks. In [26], the monitoring of the machine tool state was conducted using MTConnect. Different sensors such as RFID tags, dynamometers, accelerometers and RPM sensors were connected to Arduino or DAQ platforms and an MTConnect Adapter was used to send the sensing data to the MTConnect Agent, where all the information related to the machine tool and its corresponding sensing values are shown in real time. The XML files from the MTConnect Agent are used to send the information to the MTC application, a C# program where the data are used to show signal values in real time, generate historical data and implement intelligent algorithms. The results are then sent to the cloud, where remote monitoring and other analytics are conducted. The same authors extended their research to include a generic OPC UA information model for CNC machine tools, and an MTConnect to OPC UA interface was developed to transform the MTConnect information model and its data into their OPC UA counterparts [27].

To facilitate the implementation of TCM, the system can be simplified when dealing with repetitive machining operations [20]. Under this scenario, the machining process is conducted with the same cutting parameters and the sensor information is only related to cutting tool state. This simplification, which can be conducted after optimizing the cutting parameters and when the same product is being processed in a large or moderate volume of production, can lead to reliable and simple TCM systems with a straightforward deployment on the shop floor.

Despite the extensive research conducted on TCM, most of the previous works deal with solutions that are not straightforward to deploy on common shop floors; moreover, the sensors are invasive, the design of experiments needed to train the model requires data that should be collected and trained previously to run the system, and there is no simple way to adjust the model under different machining scenarios, e.g., a change in the tool or workpiece provider. These limitations prevent the adoption of tool condition monitoring systems by small- and medium-sized manufacturing companies. This work presents a monitoring system for tool conditions in repetitive machining operations using low-cost sensors, an IoT platform and a mobile app to show a straightforward solution with a reasonable reliability that can encourage practitioners to apply this type of systems in industry. Since the system proposed can be implemented with minimum interruption and is a low-cost solution, small- and medium-sized companies can consider it an efficient and feasible solution for cutting tool monitoring. Furthermore, the solution can be scaled to include more complex modeling tools, and the use of an IoT platform can lead to more advanced analysis of the manufacturing process. For instance, the platform could be applied for pattern search according to the feedback given by the operator, and search for problems of cutting tool breakage, built-up edge formation and so on.

This paper is organized as follows. Section 2 describes the materials and methods used for the proposed tool condition monitoring system. Section 3 shows the results of the proposed solution when implemented in a CNC lathe for face turning operations. Section 4 discusses the accuracy and cost of the proposed system and highlights the main benefits and limitations. Finally, Section 5 presents the main conclusions of the paper.

2. Materials and Methods

The proposed TCM system is composed of four parts: (i) sensor devices with edge-computing capabilities, (ii) a machine learning node, (iii) an IoT platform and (iv) a mobile app for operator feedback. All devices are connected to the local area network through Wi-Fi or Ethernet connections to ensure fast and easy deployment, and access to the Internet via a router is provided. These four main parts and the TCM operation are described in the following subsections.

2.1. Sensor Devices and Edge-Computing Node

The TCM system requires sensor devices that can indirectly monitor the state of the cutting tool. Many different sensors have been investigated and proven in the literature for TCM applications such as dynamometers, accelerometers, acoustic emission sensors, current and power sensors, and so on [28]. In this paper, the proposed TCM is based on low-cost and non-invasive devices and thus, current and sound sensors were chosen. The use of both current and sound sensors provides a solution to detect when the cutting operation is running with respect to other machine tool states such as idle or under non-cutting feed movements. Furthermore, it is assumed that tool wear has a direct impact on cutting forces and thus, the current consumption from spindle motor can be used as a sensing device for the proposed TCM system. This condition has been previously reported in the literature [14,29].

Cloud computing has recently attracted a lot of attention in manufacturing. However, latency issues prevent its implementation when dealing with time-sensitive applications, and other approaches such as fog and edge computing are more appropriate [20,30]. Therefore, the cloud server is commonly used to perform non-time-driven computing activities, such as data historic analysis, data mining and machine learning models for high-level analysis in the company, and more demanding real-time analysis is performed at the sensor or intermediate level.

In this paper, the analysis of sensor signals is conducted at the edge, using an Arduino node, and only the most relevant information from the signal is sent to the network. The sensing devices, a current clamp model SCT013 50A/1V and a sound sensor KY-038 with analog output, are installed in the machine tool. The current clamp is installed before the spindle drive that controls the spindle motor operation, and the sound sensor is placed inside the machining center at the top left corner. In order to reduce the computational burden, the root mean square (RMS) of the current signal is computed using an additional device, an RMS-to-DC converter from Mikroe based on the integrated circuit LTC1968, which sends a DC signal to the Arduino according to the RMS value of the signal using the I2C communication interface at 100 Hz. The Arduino node calculates the average of the RMS values every 10 samples and stores it. Due to the limited storage of the Arduino device, the recording is constrained to a maximum of 230 average samples, which corresponds to 23 s of cutting. On the other hand, the sound sensor signal is directly acquired at 1 kHz by the Arduino ADC. Figure 1 shows the electrical devices used at the sensing node of the proposed TCM.

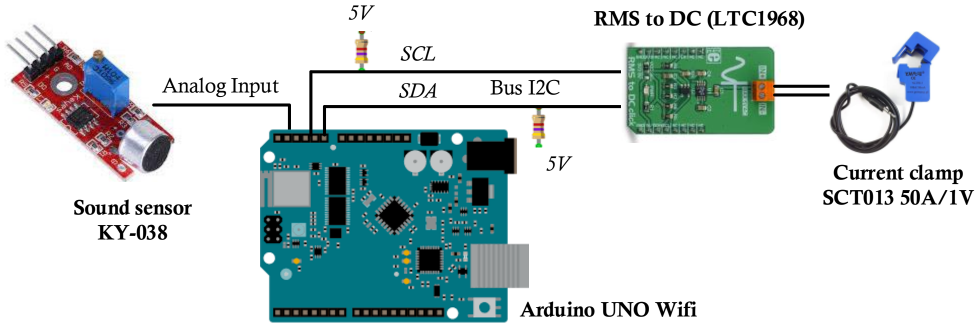


Figure 1. Sensor devices and Arduino microcontroller installed in the machine tool.

For the proposed TCM system, instead of working with the average RMS values of the signal, which is a common descriptor used in monitoring systems [28], the Arduino node provides a more sophisticated descriptor, a similarity index. In the literature, more than 60 different similarity measures have been proposed to quantify how alike two objects/signals are [31]. Popular indices are distance-based and angle-based similarity measures. In this paper, the similarity index is evaluated by the Arduino node; thus, a simple index with low computational cost is required. A simple index that has been successfully applied in TCM systems is reported in [20,32]. In these investigations, the authors proposed a tool wear monitoring system based on a similarity analysis between a reference signal and signals under normal operation. The calculation of the similarity index is conducted with the following equation:

$$f(k) = \frac{1}{n}\sum_{i=0}^{n}\left|\frac{h(i) - g(i+k)}{h(i)}\right|$$

where $f(k)$ is the average difference between the reference signal (new tool), denoted as $g(i)$, and the signal that is being analyzed, denoted as $h(i)$, where i denotes one out of n data points from both signals. The shift k is used in case the reference signal is shorter that the analyzed signal and there is no initial value to start the signal comparison. In those cases, the shift k is used to move the reference signal and find the instant where the similarity of both signals is the maximum. This may be of interest to compare only the steady part of the signal and avoid transient moments such as cutting entering movements. Figure 2 illustrates this comparison process.

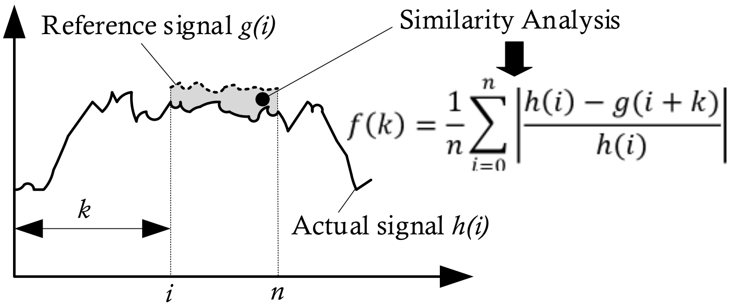

Reference signal *g(i)*
Similarity Analysis
$f(k) = \frac{1}{n}\sum_{i=0}^{n}\left|\frac{h(i) - g(i+k)}{h(i)}\right|$
k
Actual signal *h(i)*
i
n

Figure 2. Similarity analysis between reference signal and actual signal.

In this paper, we use a similar approach, and we give the Arduino node the capability of acquiring a reference signal, storing it as a reference, and comparing the reference with every signal after the cutting operation to calculate the similarity index. These similarity indices are then sent to the network for training, modeling and running the TCM system. In practice, the similarity index is defined by the lowest discrepancy between the two signals (i.e., maximum alignment), and ranges from 100%, which means an identical signal, to 0%. Mathematically, the similarity index is computed as follows:

$$Similarity\ Index = e^{-f(k)_{min}}$$

In this paper, the similarity analysis is slightly modified due to computation constraints in the Arduino node. Thus, the shift is removed to conduct the comparison when the sound signal is activated and the current value exceeds a threshold, and the similarity index is used without the exponential calculation, which means that the value applied for the similarity is $f(0)$.

2.2. Machine Learning Node

The sensing node presents a limited computing capability, and its application is restricted to extract descriptors from raw signal data. In the proposed TCM, the Arduino node evaluates the similarity index between the reference signal and the cutting signal. For modeling purposes, an additional node with higher capabilities is introduced in the TCM. This node, named the machine learning (ML) node, is composed of a BeagleBone Black (BBB) board. A BBB is a low-cost board based on Linux with a AM335x 1 GHz ARM® Cortex-A8 processor, 512 MB DDR3 RAM and 4 GB 8-bit eMMC on-board flash storage and Ethernet connectivity. The main goal of this node is to run an ML algorithm for predicting the tool wear state of the cutting tool. Python 3.7 is installed in the BBB which facilitates the development of ML algorithms thanks to libraries such as scikit-learn. Using this library, the implementation of common ML algorithms such as support vector machines, decision trees, Naïve Bayes classifiers, k-Nearest neighbors, etc., is fast and easy.

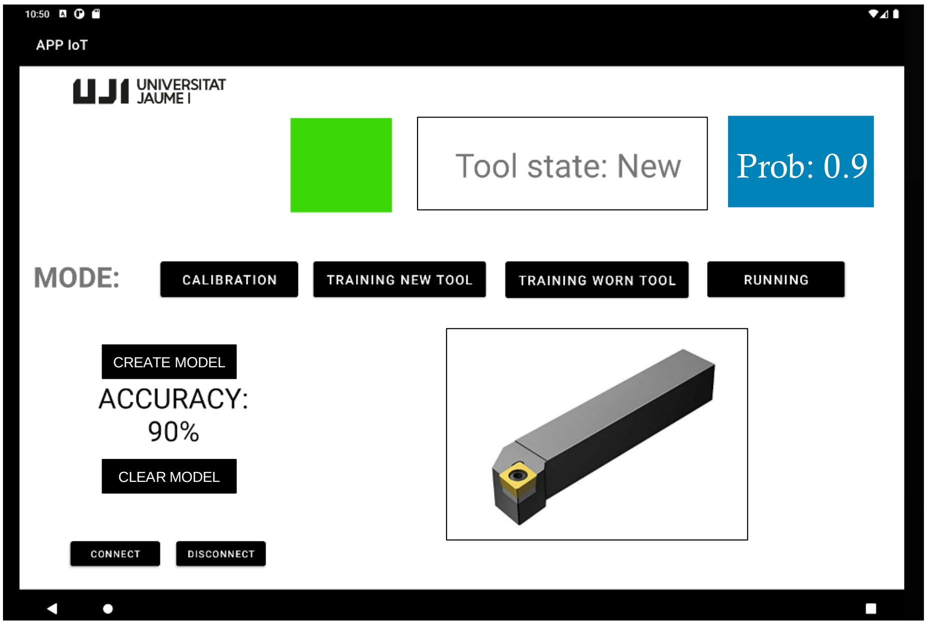
In this paper, we propose the use of a more straightforward and easy classification model, the logistic regression model. Logistic regression is a relatively uncomplicated linear classifier that works well when dealing with non-complex problems. Since we are interested in testing the platform, and as the relationship between current/power consumption and tool wear is well known, we apply this algorithm for the sake of simplicity. Before the logistic regression, the data are normalized using the min-max normalization. Note that any other ML algorithm could be applied with the inclusion of the proper code into the Python script.

2.3. Mobile App

To ensure the correct communication with the operator, a mobile app is implemented using Android Studio. The purpose of the app is to facilitate the deployment of the system and avoid the use of any additional software at any configuration stage of the TCM system (i.e., calibration, training, modeling and running). The app will be used for calibrating, training and running the TCM system and it will show the estimated cutting tool state after every cutting operation. Several buttons are included in the app to change the operation mode of the TCM system. These operation modes are:

- Calibration Mode. This mode is used to prepare the Arduino for recording the current signal as the reference signal for the process.
- Training Mode. Under this mode, a set of experiments can be conducted to create a data set for building an ML model. The training mode is available in two ways: "training new tool" and "training worn tool". This labeling is needed to create the training database for classification.
- Modeling Mode. This mode sends a message to the machine learning node to build the ML model and it is executed by pressing the button "create model". If the model accuracy is not adequate, the operator can restart the modeling and training data with the "clear model" button or add additional training data.
- Running Mode. If the model is approved by the operator, the TCM can start being used under normal operations, which is called the running mode. Under this mode, the app shows the estimated tool state (i.e., new or worn state) and its probability on the screen after every cutting operation.

Figure 3 shows the main window of the app where the menu buttons are and where the actual state of the cutting tool is reported.



Tool state: New

Prob: 0.9

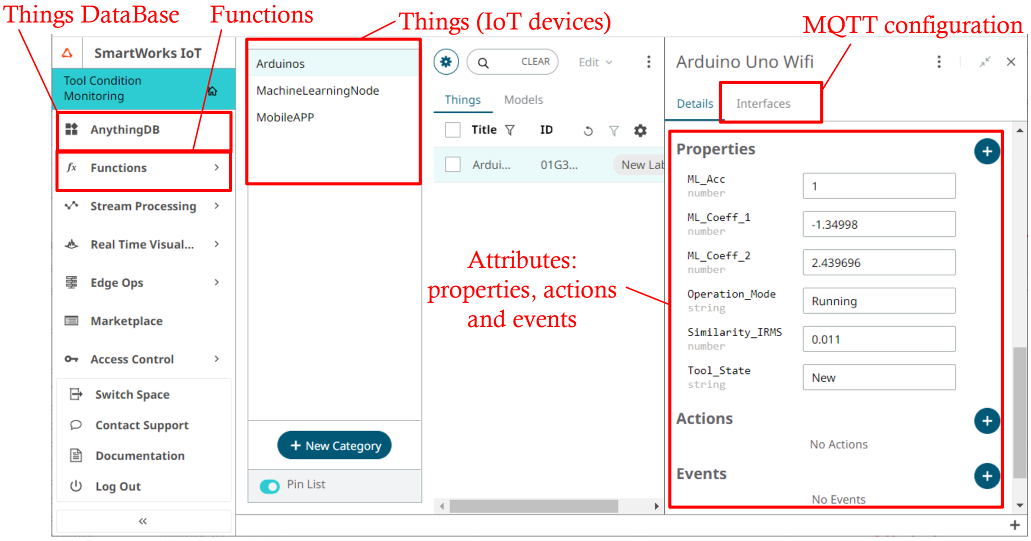
ACCURACY: 90%

Figure 3. Main window of the developed app for the operator's interaction and the IoT platform.

2.4. IoT Platform

The third part of the TCM system consists of the IoT platform. The IoT platform used is Altair SmartWorks, a powerful tool to rapidly connect IoT devices, manage access to devices and generate scripts to execute actions. This platform was selected at the researchers' convenience, but any other IoT platform based on an open source could have been selected at no cost, such as Things Board or Open Remote.

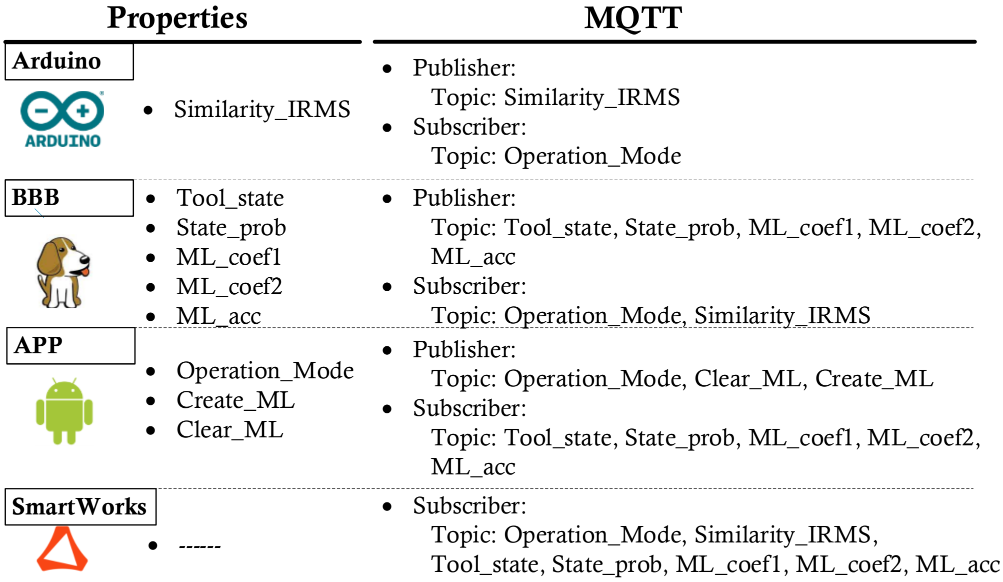
The SmartWorks platform is composed of AnythingDB, a database to store the data from the registered IoT devices; functions that let users develop custom code in GO and Python to implement machine learning models; and real-time visualization tools to visualize dashboards with all the desired information. Figure 4 shows the main panel for setting up IoT devices in SmartWorks and some characteristics.



ML_Acc (number): 1
ML_Coeff_1 (number): -1.34998
ML_Coeff_2 (number): 2.439696
Operation_Mode (string): Running
Similarity_IRMS (number): 0.011
Tool_State (string): New

Figure 4. Main panel of SmartWorks, where AnythingDB is shown together with the properties of the Thing Arduino node.

The IoT devices are defined according to the standard definition W3C and their descriptions are encoded in JSON format. The attributes for each device are properties, actions and events. Properties refer to values that the devices are reporting, actions refer to how we can interact with the device and events refer to the definition of specific states of the process that want to be notified. For the proposed TCM system, the properties defined for each IoT device are shown in Figure 5.



	Properties	MQTT
Arduino	• Similarity_IRMS	• Publisher: Topic: Similarity_IRMS • Subscriber: Topic: Operation_Mode
BBB	• Tool_state • State_prob • ML_coef1 • ML_coef2 • ML_acc	• Publisher: Topic: Tool_state, State_prob, ML_coef1, ML_coef2, ML_acc • Subscriber: Topic: Operation_Mode, Similarity_IRMS
APP	• Operation_Mode • Create_ML • Clear_ML	• Publisher: Topic: Operation_Mode, Clear_ML, Create_ML • Subscriber: Topic: Tool_state, State_prob, ML_coef1, ML_coef2, ML_acc
SmartWorks	• ------	• Subscriber: Topic: Operation_Mode, Similarity_IRMS, Tool_state, State_prob, ML_coef1, ML_coef2, ML_acc

Figure 5. Properties for the IoT devices defined in SmartWorks and MQTT topics, publishers and subscribers.

The communication between all devices is carried out by the MQTT protocol. MQTT is a standard for IoT messaging and it allows for messaging from device to cloud and cloud to device. Client connections are always handled by an MQTT broker, which is responsible for receiving and filtering the messages, determining who is subscribed to each message and sending them. Three main roles can be found in the MQTT protocol: the publisher, the broker and the subscriber. The publisher publishes messages to a topic on an MQTT broker, which acts as a server and collects the information from all publishers and distributes them to the subscribers, which have been previously subscribed to a certain topic. For the proposed TCM system, we use the MQTT broker that is provided by Altair SmartWorks Platform, and the MQTT credentials and topics are defined in the device creation/modification forms. Figure 5 shows the topics defined for the proposed TCM system with their publishers and subscribers.

2.5. System Operation

To set up the TCM system, the following 4-step procedure is required. Firstly, the operator mounts a new cutting tool insert and presses the button in the app for calibration. The operator executes the machining operation and the Arduino saves the values obtained from the RMS to DC converter through the I2C bus. The recording starts when the sound detects the cutting operation. The Arduino saves these RMS values as a reference profile.

Secondly, a set of experiments is conducted to create the data set needed for building the ML model. Before each machining operation, the operator presses the button "training new tool" or "training worn tool" depending on the state of the tool mounted in the tool holder. The Arduino saves the data during the cutting and after finishing, the recorded profile is compared with the reference profile by the evaluation of the similarity index (SI). Then, the Arduino sends the SI to the network using the MQTT protocol, and both the IoT platform and the ML node receive and store the information. When the set of experiments is finished, the operator presses the button "Create Model" from the app. At this point, the ML node receives the message and a Python script is executed to build a logistic regression model based on the SI indices registered previously.

In a third step, the built model should be approved. After the ML node has built the model, the results in terms of accuracy and the regression coefficients are sent to the IoT platform, and the accuracy of the trained model is shown in the app for the operator's validation. The accuracy refers to the accuracy obtained by the ML model (in our case, the logistic regression model) using the training data. If the accuracy is not adequate, the operator can add additional training data using "training new tool" or "training worn tool". By pressing the "create model" button again, the model is retrained. If the operator wants to remove the training data, the "clear model" button is pressed instead. In case the accuracy is adequate for the application, the TCM is ready for running.

After the proper model is built, the last step refers to the operation of the TCM. The operator presses the running mode button, and after each machining operation, the TCM will report on the app the estimated state of the cutting tool and its probability. During this stage, the Arduino will send the SI value to the network, and the ML node will respond to that value with the estimated tool state and probability according to the previously learnt model. All the information during the running mode is stored at the IoT for future analysis. Figure 6 shows an overview of the operation of the TCM system.

2.6. Experimental Setup for TCM Validation

The proposed TCM system was tested in the laboratory environment shown in Figure 7. The equipment used for the validation included a CNC lathe (Lealde tcn-10 and Fagor 8055T control), a Ø130 mm round bar as workpiece material (unalloyed steel UNE F114, equivalent to ISO C45 carbon steel) and coated cutting inserts with ISO code CNMG120408. The machining operation analyzed was a face turning operation with the following cutting parameters: feed rate 0.1 mm/rev, cutting speed 250 m/min and depth of cut 2 mm. The operation was conducted with constant cutting speed and the maximum spindle speed was 2000 rpm.

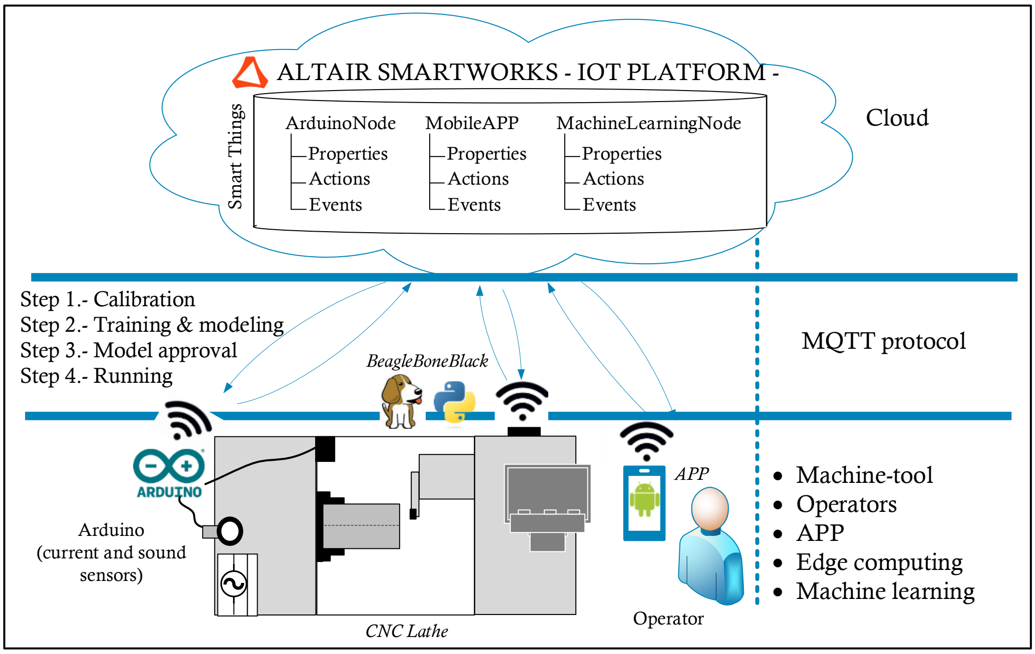


Figure 6. Overview of the proposed tool condition monitoring system.

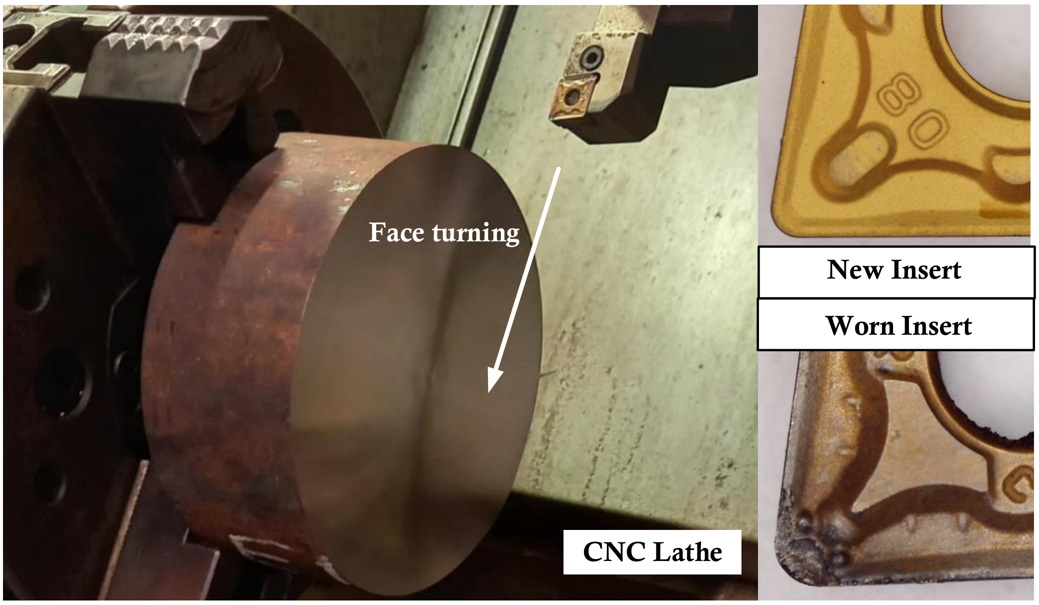


Figure 7. CNC lathe, workpiece and cutting tool applied in the implementation of the TCM system.

The experimental results can be divided into four parts. First, an experiment was conducted to analyze the feasibility of using the sound sensor and the current clamp for the TCM purpose. Secondly, a calibration step was conducted to record the reference signal that will be used for the SI evaluation. Thirdly, a design of experiments (DoE) was conducted to test the capability of the IoT platform to train and model an ML model, i.e., the logistic regression model. To this end, six face turning operations were conducted, three with new tools (or low wear value) and three with tools that are considered totally worn and require replacement. Finally, the TCM system was validated in the running stage, where a set of cutting experiments were conducted and the classification results were compared with the real ones. To this end, 17 face turning operations with 8 new tools and 9 worn tools were used.

3. Results

3.1. Sound and Current Signals

In order to prove that the current and sound sensor can provide enough information for TCM purposes, a short experiment was conducted. The experiment consisted of acquiring both signals under a normal cutting operation.

To analyze the sound signal, a face turning operation was conducted and the sound signal was acquired at a 1 kHz sampling rate. The raw data obtained from the sensor are shown in Figure 8a. The use of this raw signal as an indicator of the start of the cutting operation is not adequate, and to overcome this limitation, the signal is processed through the Arduino node to evaluate the average peak to peak value from every 50 samples. The resulting signal is presented in Figure 8b, which shows a filtered signal that can be used as a reliable measure to identify the different states of the machine tool (idle, cutting entry, cutting state).

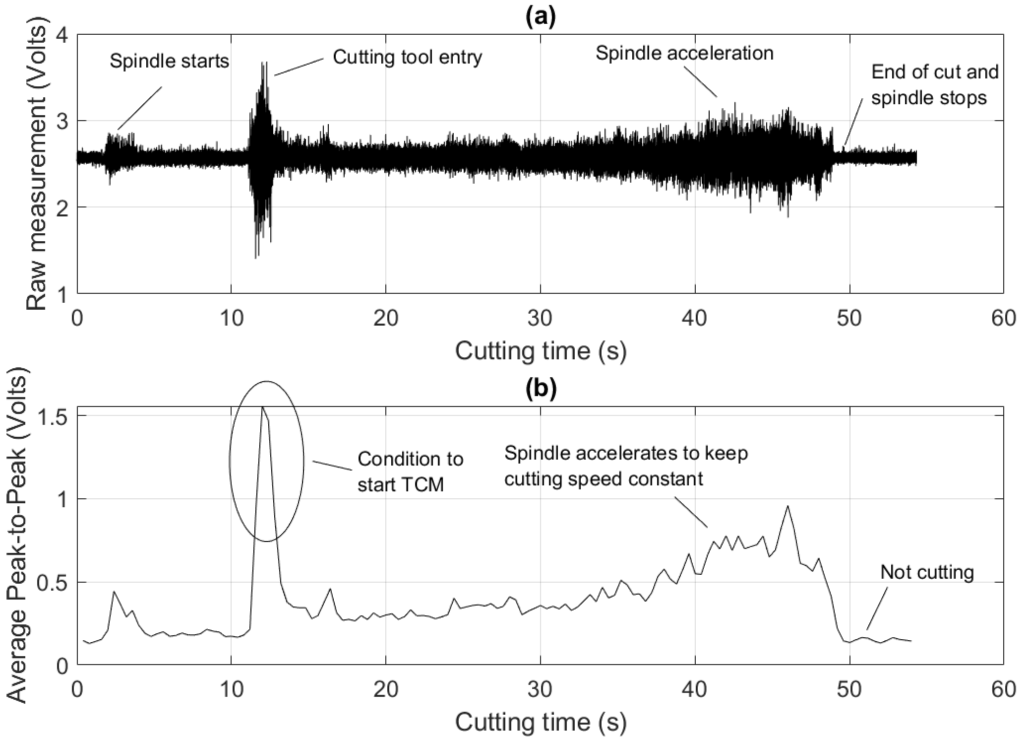


Figure 8. Sound signal acquired during machining and identification of machine tool states: (**a**) raw measurement and (**b**) average peak to peak value from every 50 samples.

On the other hand, the current signal is acquired by the RMS to DC converter and read from the I2C bus at a frequency of 100 Hz. The Arduino node calculates the average of the RMS value every 10 samples and stores it as the current consumption. Figure 9 shows the results of the current signal acquired in a face turning operation and the different states of the spindle and cutting tool.

In order to analyze the potential use of the current signal profile as an indicator of tool wear, four face turning operations were conducted. The first two operations were carried out with two different new cutting tools. The third operation was conducted with a worn tool that needs replacement. In fact, the operation was stopped to avoid tool breakage (excessive chip and heat was generated). A final operation was conducted without cutting to compare the current consumption due to spindle and axis movements. From the results, shown in Figure 10, we can state that a high increase in current consumption exists when the cutting tool is in an extremely worn condition, with the existence of slight differences between current consumption in new cutting tools and a potential distinction between cutting with new or worn tools and non-cutting situations using the current clamp, which proves the use of this non-invasive, low-cost sensor as a suitable option for low-cost TCM systems.

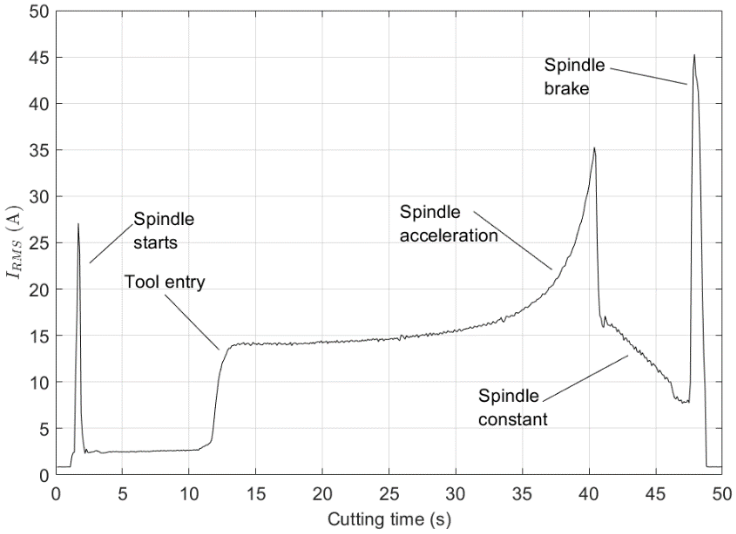


Figure 9. Profile acquired from the current signal after the RMS to DC converter.

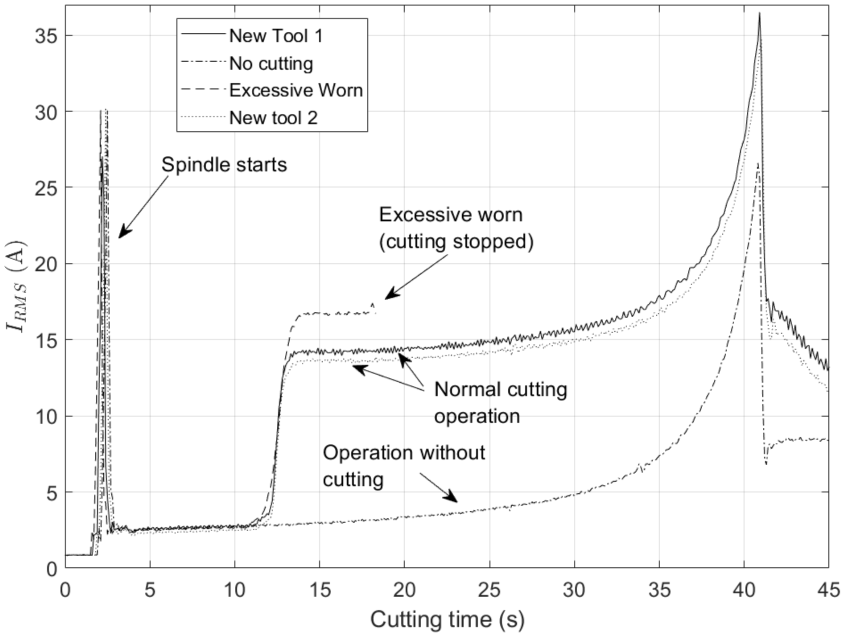


Figure 10. Current signal profile for a face turning operation with a different tools and conditions.

3.2. Calibration Stage

For the calibration stage, the operator sends the calibration mode via the mobile app and a cutting operation with a new tool is conducted. The Arduino starts recording the data from the current clamp via I2C when the sound sensor detects the beginning of the cutting process. The maximum length of the vector due to memory constraints is fixed to 230 samples and thus, if the sound sensor does not indicate that the cutting process has finished, the recording is stopped when 230 samples (i.e., 23 s) are registered. Figure 11 shows the recorded signal that is used as a reference signal for the SI analysis together with the entire current signal after the sound sensor has indicated the beginning of the cutting process.

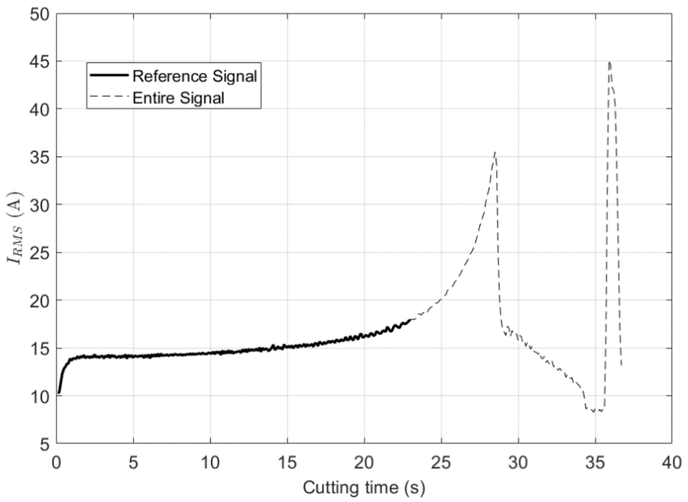


Figure 11. Reference signal after calibration to be used for the similarity index.

3.3. Machine Learning Model (Training and Modeling Stage)

In order to test the capability of the TCM system, a set of cutting operations with new and worn cutting tools were conducted. After each cutting operation, the Arduino node sends the similarity index to the network, which is registered as training data with the label of "new" or "worn" previously marked in the app by the operator. Table 1 shows the experiments conducted and the similarity indices sent by the Arduino node to the network. Note that in this training set, a worn tool is one that presents a high level of wear according to the operator, and a new tool is any tool that could be still used in the cutting process. Therefore, there is no need to measure the flank wear (the International Organization for Standardization refers to 0.3 mm as the maximum value of the flank wear to consider the tool as worn [33]), since our purpose is to provide a fast and low-cost TCM system, and in many industrial practices, the cutting tool is changed when the operator considers it to be worn, usually due to abnormal cutting sound, low surface quality, chip color and so on. To cover different cutting situations, both new and worn inserts with a low value of wear (i.e., tools that can still be used) were used as "new" tools.

Table 1. Machining operations conducted for training the ML model.

Experiment	Tool State	Similarity Indices	Normalized Indices
1	Worn	0.0585	0.731
2	New	0.0289	0.143
3	Worn	0.0426	0.415
4	New	0.0217	0
5	Worn	0.072	1
6	New	0.049	0.543

When the set of experiments is finished, the operator uses the mobile app to request the creation of the ML model. At this point, the BBB board runs the Python code to build an ML model based on logistic regression. Before the logistic regression, the data are normalized using the min-max normalization, as shown in Table 1. For illustrative purposes, Figure 12 shows the resulting logistic regression of the training data and the expected similarity threshold if the decision of the state of the tool is carried out under a probability of 0.5. Note that the model is trained only with six data points in order to show the fast deployment capability of the proposed TCM. However, larger data sets that are representative of the process are highly recommended to provide a better classifier, although it may depend on the application.

After training the model, the accuracy is sent back to the app to be validated by the operator. For this application, the reported accuracy was 83%. In case the accuracy is not adequate, the operator can obtain additional experimental data to augment the training data set. The operator can also delete the previous data stored in the machine learning node and repeat the training process.

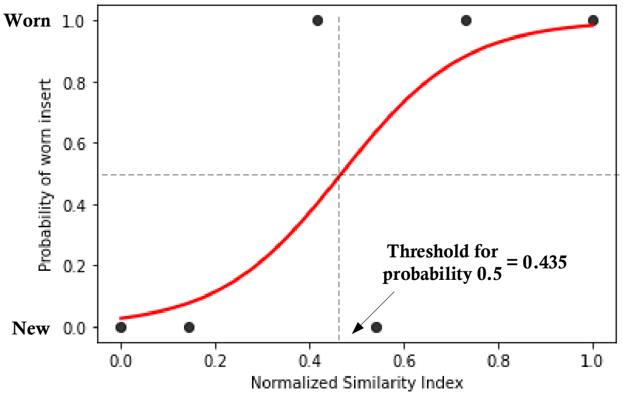


Figure 12. Logistic regression (red curve) obtained from the training data (black dots) and threshold of the normalized similarity index for a probability of 0.5.

It should be remarked that for practical purposes, the set of experiments for training is defined, but for industrial applications, this set of experiments could be the same cutting operations that are currently going on at the shop floor level, and only the operator's feedback through the app is required to use this information as training data.

3.4. Classification Results from the Machine Learning Model (Running Stage)

After training the model at the BBB board, the TCM is ready to operate. To test the performance of the TCM system, a set of additional cutting experiments were conducted with new and worn tools. The TCM system estimates the cutting state after each cutting operation, and this estimation together with its probability are shown in the app to visually inform the operator. Table 2 shows the cutting operations for validating the TCM system together with the probability of the prediction given by the ML model from the BBB board. From the experimental validation, 15 out of 17 cases were correctly classified, which indicates a validation accuracy of 88%. In two experiments, experiments 5 and 16, the new tool was classified as a worn tool according to the ML model. Experiment 16 predicted a worn tool with a probability of 0.56, but the actual state was a new tool. Experiment 5 predicted a worn tool with a probability of 0.82 but the actual state was a new tool.

Table 2. Predicted tool state and probability of the predicted state.

Experiment	Actual Tool State	Predicted Tool State	Probability
1	New	New	0.60
2	Worn	Worn	0.59
3	New	New	0.51
4	Worn	Worn	0.81
5	New	Worn	0.82
6	Worn	Worn	0.75
7	New	New	0.80
8	Worn	Worn	0.82
9	New	New	0.75
10	Worn	Worn	0.99
11	New	New	0.73
12	Worn	Worn	0.69
13	Worn	Worn	0.58
14	New	New	0.89
15	Worn	Worn	0.6
16	New	Worn	0.56
17	Worn	Worn	0.83

4. Discussion

The experimental results have validated the proposed TCM system. The use of an Arduino board as an edge-computing node was proved to be sufficient for recording reference signals and computing similarity indices. As it is shown in the training and modeling steps, the use of the similarity index serves as an adequate descriptor of the state of the cutting tool. Unlike considering only the RMS signal value from the whole signal, the similarity index compares the RMS values from a reference signal with respect to the actual signal, and it can provide more reliable information about the increase in the current consumption along the cutting profile. For instance, one may consider cutting operations where tool load varies along the trajectory.

The ML algorithm applied is based on logistic regression and it is built in the BBB board using Python and the scikit-learn library. This machine learning algorithm was applied for the sake of simplicity, but more complex algorithms can be applied using the same library. However, an increase in the complexity of the model may require a larger set of experimental data. For repetitive machining operations, the logistic regression algorithm may provide an acceptable performance without the need for advanced models. In fact, only 7 cutting operations for calibration and training (1 as a reference profile and 6 for training purposes) and 17 cutting operations for validating were necessary to set up and validate the TCM system. The training model presented an accuracy of 83%, which was considered adequate at this stage. However, the advantage of this TCM is that, in case the accuracy of the model is not adequate, the system can be easily retrained by adding supplementary training data. For this purpose, the operator only needs to conduct an additional experiment and introduce the new data using the app.

The TCM was validated with 17 cutting operations, and 15 out of 17 cases were correctly classified. In two cases, two new cutting tools were considered as worn, which is due to the variability in the current consumption among cutting tools. A training stage with more experimental data may be able to provide better generalization and improve the tool wear classification. Note that in this work, we have minimized the amount of experimental training data to highlight the potential fast deployment with reasonable performance of the proposed TCM system.

From an economical point of view, the proposed TCM system is a low-cost system with a global cost of less than EUR 170, excluding the tablet/mobile device for running the app. In more detail, the costs are defined as follows: BBB board (EUR 70), Arduino UNO Wi-Fi rev2 board (EUR 45), current clamp sensor SCT013 50A/1V (EUR 11), sound sensor KY-038 (EUR 2) and RMS-to-DC converter MiKroe (EUR 35). The deployment of the TCM is fast and easy since there is no invasive sensor. The sound sensor is placed with a magnet or something similar in the top left corner of the machine tool, and the current clamp is installed in the electrical panel of the machine tool. The BBB board, the RMS-to-DC converter and the Arduino board can be fixed to the electrical panel using different magnets.

Three main limitations can be identified in the proposed TCM system. First, the system is used to assist the operator, and the information displayed on the mobile app is informative. The automatic tool change is not included in the system since this requires communication with the CNC, and in many cases, this step requires open controls or specific characteristics of the CNC. Second, the application is restricted to repetitive machining operations, where cutting trajectory and cutting conditions are fixed. Then, the main variations in the current signal are due to the wear state of the cutting tool. Under this scenario, a fast deployment of the TCM system is possible. For more flexible operations where tool paths and cutting conditions change, the ML model must include more variables and the experiment should be enough to acquire the interaction between all factors. Thus, the deployment would increase notably and a more detailed analysis of signal descriptors for modeling is needed. However, the same structure of the TCM could be kept with an increase in the experimental part. Third, the reliability of the proposed TCM is directly related to the influence of tool wear on the current consumption. If cutting conditions are not very demanding and the cutting forces are low, the increase in the current consumption due to tool wear may be negligible and thus, this TCM system cannot be applied.

The presented work can be extended to include the automatic tool replacement of the machine tool. For this purpose, open CNC controllers or controllers with OPC UA server can be used for a straightforward implementation. Legacy machine tools and non-open controllers can apply MTConnect if MT adapters are available [34]. Furthermore, the proposed solution uses an ML algorithm based on logistic regression. Further investigation can be conducted to explore other ML algorithms during the cutting tool life cycle in order to obtain models that can be adapted when the initial conditions change, such as different cutting tool vendors, workpiece hardness changes and so on.

5. Conclusions

This paper presents a tool condition monitoring (TCM) system for fast deployment based on an IoT platform and low-cost sensors with edge-computing capabilities. The TCM is composed of an Arduino node to acquire sound and current consumption signals through low-cost and non-invasive sensors; a machine learning node based on a BeagleBoneBlack unit to build the tool wear model using a logistic regression classifier; and an IoT platform to provide the IoT network for plug and play practices, and to provide a future option for developing more advanced models using historical data. As a signal descriptor for modeling purposes, a straightforward similarity index with low computational cost was used at the Arduino node.

Instead of other more advanced TCMs that are time-consuming and require expensive sensor systems, the proposed TCM is able to identify if the tool is worn with an accuracy of 88% under a system that only requires an investment around EUR 170 and a few hours for deployment. In fact, for the proposed application in face turning operations, only seven runs were needed to build the model, although more runs may be recommended at the training stage for better generalization. The integration of the TCM with a mobile app and the use of a machine learning classifier allows the operator to easily rebuild the model and adapt the TCM to changes in the operation, tool, workpiece and so on. Furthermore, the potential installation of this TCM is possible with almost no interruption at the shop floor level.

As an additional remark, the proposed TCM is applied as an assistance tool to operators, and it can provide a fast and low-cost solution to decide a proper tool change. The system as it is presented is not intended to be applied to unmanned machining systems. In those cases, the use of sensor fusion systems with a larger number of descriptors from sensor signals and more complex classification models may be required. However, our proposed TCM can be easily upgraded to derive more complex models including more sensors in the network and using the IoT platform where more advanced models through machine learning techniques can be applied.

Author Contributions: Conceptualization, J.V.A.-N. and P.R.C.; Funding acquisition, J.V.A.-N. and P.R.C.; Investigation, J.M.F.; Methodology, J.V.A.-N.; Project administration, J.V.A.-N. and P.R.C.; Software, J.M.F.; Supervision, F.R.S.; Validation, J.V.A.-N.; Visualization, J.M.F.; Writing—original draft, J.V.A.-N.; Writing—review and editing, S.B.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by UNIVERSITAT JAUME I, grant number UJI-B2020-33 and UJI-B2022-49.

Data Availability Statement: The data presented in this study are available on request to the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Davim, J.P. *Sustainable Machining*; Springer: Berlin/Heidelberg, Germany, 2017.
2. Jain, A.K.; Lad, B.K. A Novel Integrated Tool Condition Monitoring System. *J. Intell. Manuf.* **2019**, *30*, 1423–1436. [CrossRef]
3. Astakhov, V.P. Improving Sustainability of Machining Operation as a System Endeavor. In *Sustainable Machining*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 1–29.
4. Salonitis, K.; Kolios, A. Reliability Assessment of Cutting Tool Life Based on Surrogate Approximation Methods. *Int. J. Adv. Manuf. Technol.* **2014**, *71*, 1197–1208. [CrossRef]
5. Drouillet, C.; Karandikar, J.; Nath, C.; Journeaux, A.C.; el Mansori, M.; Kurfess, T. Tool Life Predictions in Milling Using Spindle Power with the Neural Network Technique. *J. Manuf. Process.* **2016**, *22*, 161–168. [CrossRef]
6. Martinova, L.I.; Grigoryev, A.S.; Sokolov, S.V. Diagnostics and Forecasting of Cutting Tool Wear at CNC Machines. *Autom. Remote Control.* **2012**, *73*, 742–749. [CrossRef]
7. Ranjan, P.; Hiremath, S.S. Role of Textured Tool in Improving Machining Performance: A Review. *J. Manuf. Process.* **2019**, *43*, 47–73. [CrossRef]
8. Zindani, D.; Kumar, K. A Brief Review on Cryogenics in Machining Process. *SN Appl. Sci.* **2020**, *2*, 1–10. [CrossRef]
9. Al-Ghamdi, K.A.; Iqbal, A. A Sustainability Comparison between Conventional and High-Speed Machining. *J. Clean. Prod.* **2015**, *108*, 192–206. [CrossRef]
10. Mohanraj, T.; Shankar, S.; Rajasekar, R.; Sakthivel, N.R.; Pramanik, A. Tool Condition Monitoring Techniques in Milling Process-a Review. *J. Mater. Res. Technol.* **2020**, *9*, 1032–1042. [CrossRef]
11. Serin, G.; Sener, B.; Ozbayoglu, A.M.; Unver, H.O. Review of Tool Condition Monitoring in Machining and Opportunities for Deep Learning. *Int. J. Adv. Manuf. Technol.* **2020**, *109*, 953–974. [CrossRef]
12. Kuntoğlu, M.; Aslan, A.; Pimenov, D.Y.; Usca, Ü.A.; Salur, E.; Gupta, M.K.; Mikolajczyk, T.; Giasin, K.; Kapłonek, W.; Sharma, S. A Review of Indirect Tool Condition Monitoring Systems and Decision-making Methods in Turning: Critical Analysis and Trends. *Sensors* **2021**, *21*, 108. [CrossRef]
13. Nasir, V.; Sassani, F. A review on deep learning in machining and tool monitoring: Methods, opportunities, and challenges. *Int. J. Adv. Manuf. Technol.* **2021**, *115*, 2683–2709. [CrossRef]
14. Zhang, X.Y.; Lu, X.; Wang, S.; Wang, W.; Li, W.D. A Multi-Sensor Based Online Tool Condition Monitoring System for Milling Process. In *Procedia CIRP*; Elsevier B.V.: Amsterdam, The Netherlands, 2018; Volume 72, pp. 1136–1141.
15. Teti, R. Advanced IT Methods of Signal Processing and Decision Making for Zero Defect Manufacturing in Machining. In *Procedia CIRP*; Elsevier B.V.: Amsterdam, The Netherlands, 2015; Volume 28, pp. 3–15.
16. Ahmad, M.I.; Saif, Y.; Yusof, Y.; Daud, M.E.; Latif, K.; Kadir, A.Z.A. A Case Study: Monitoring and Inspection Based on IoT for Milling Process. *Int. J. Adv. Manuf. Technol.* **2020**, *118*, 1305–1315. [CrossRef]
17. Wu, D.; Jennings, C.; Terpenny, J.; Kumara, S.; Gao, R.X. Cloud-Based Parallel Machine Learning for Tool Wear Prediction. *J. Manuf. Sci. Eng. Trans. ASME* **2018**, *140*, 041005. [CrossRef]
18. Župerl, U.; Stepien, K.; Munđar, G.; Kovačič, M. A Cloud-Based System for the Optical Monitoring of Tool Conditions during Milling through the Detection of Chip Surface Size and Identification of Cutting Force Trends. *Processes* **2022**, *10*, 671. [CrossRef]
19. Wu, D.; Liu, S.; Zhang, L.; Terpenny, J.; Gao, R.X.; Kurfess, T.; Guzzo, J.A. A Fog Computing-Based Framework for Process Monitoring and Prognosis in Cyber-Manufacturing. *J. Manuf. Syst.* **2017**, *43*, 25–34. [CrossRef]
20. Liu, R. An Edge-Based Algorithm for Tool Wear Monitoring in Repetitive Milling Processes. *J. Intell. Manuf.* **2022**, 1–11. [CrossRef]
21. Kondo, Y.; Yamaguchi, M.; Sakamoto, S.; Yamaguchi, K. A Study on Cyber-Physical System Architecture to Predict Cutting Tool Condition in Machining. *Int. J. Mech. Eng. Robot. Res.* **2020**, *9*, 565–569. [CrossRef]
22. Park, D.; Kim, S.; An, Y.; Jung, J.Y. Lired: A Light-Weight Real-Time Fault Detection System for Edge Computing Using LSTM Recurrent Neural Networks. *Sensors* **2018**, *18*, 2110. [CrossRef]
23. MTConnect Institution. Available online: https://www.mtconnect.org/ (accessed on 15 November 2022).
24. OPC Foundation. Available online: https://opcfoundation.org/ (accessed on 15 November 2022).
25. de Farias, A.; Luiz, S.; de Almeida, R.; Delijaicov, S.; Seriacopi, V.; Bordinassi, E.C. Simple Machine Learning Allied with Data-Driven Methods for Monitoring Tool Wear in Machining Processes. *Int. J. Adv. Manuf. Technol.* **2020**, *109*, 2491–2501. [CrossRef]
26. Liu, C.; Xu, X.; Peng, Q.; Zhou, Z. MTConnect-Based Cyber-Physical Machine Tool: A Case Study. In *Procedia CIRP*; Elsevier B.V.: Amsterdam, The Netherlands, 2018; Volume 72, pp. 492–497.
27. Liu, C.; Vengayil, H.; Lu, Y.; Xu, X. A Cyber-Physical Machine Tools Platform Using OPC UA and MTConnect. *J. Manuf. Syst.* **2019**, *51*, 61–74. [CrossRef]
28. Abellan-Nebot, J.V.; Romero Subirón, F. A Review of Machining Monitoring Systems Based on Artificial Intelligence Process Models. *Int. J. Adv. Manuf. Technol.* **2010**, *47*, 237–257. [CrossRef]
29. Moliner-Heredia, R.; Peñarrocha-Alós, I.; Abellán-Nebot, J.V. Model-Based Tool Condition Prognosis Using Power Consumption and Scarce Surface Roughness Measurements. *J. Manuf. Syst.* **2021**, *61*, 311–325. [CrossRef]
30. Pizoń, J.; Lipski, J. Perspectives for Fog Computing in Manufacturing. *Appl. Comput. Sci.* **2016**, *12*, 37–46.
31. Xing, K.; Achiche, S.; Mayer, J.R.R. Five-Axis Machine Tools Accuracy Condition Monitoring Based on Volumetric Errors and Vector Similarity Measures. *Int. J. Mach. Tools Manuf.* **2019**, *138*, 80–93. [CrossRef]

32. Liu, R.; Kothuru, A.; Zhang, S. Calibration-Based Tool Condition Monitoring for Repetitive Machining Operations. *J. Manuf. Syst.* **2020**, *54*, 285–293. [CrossRef]
33. *ISO 3685:1993*; Tool-Life Testing with single-Point Turning Tools. International Standard, ISO: Geneva, Switzerland, November 1993.
34. Edrington, B.; Zhao, B.; Hansel, A.; Mori, M.; Fujishima, M. Machine Monitoring System Based on MTConnect Technology. In *Procedia CIRP*; Elsevier B.V.: Amsterdam, The Netherlands, 2014; Volume 22, pp. 92–97.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.