



# Automatic classification of smart sensor data for evaluating machine tool efficiency

Marco Sortino<sup>1</sup> · Emanuele Vaglio<sup>1</sup>

Received: 5 December 2023 / Accepted: 24 April 2024  
© The Author(s) 2024

## Abstract

The assessment of production plant efficiency is crucial for optimizing the operational performance of manufacturing systems. In traditional facilities, automated data collection is limited and information primarily relies on operators declarations, which are prone to inaccuracy. There is therefore a need for readily accessible digital alternatives. This paper introduces a cost-effective method for classifying the status of machine tools using smart sensors to monitor their primary doors with minimal integration, and a streamlined algorithm for efficient data processing. The innovative algorithm was conceived using data collected in over 3 months in a manufacturing plant comprising 50 diverse machine tools engaged in batch production for the automotive industry, and is based on non-dimensional thresholds, making it suitable for generic applications requiring classification of repetitive patterns. Also, a realistic simulator was developed to provide reliable data for algorithm accuracy evaluation. The classification performance was fully tested using synthetic data, showing very good accuracy. In addition, the performance of the algorithm was compared to basic machine learning approaches further proving the validity of the proposed method. Ultimately, the classification algorithm was employed to assess the Overall Equipment Effectiveness (OEE) of the real plant machines, which were closely aligned with the estimates provided by the enterprise management.

**Keywords** Machine tool · Monitoring · Status classification · Smart sensors · Efficiency

## Nomenclature

$BA$	Balanced accuracy	$d_{ts1,ts2,str}$	Straight difference between time series
$C$	Number of cases	$d_{z,w}$	Difference between cumulative sum arrays
$c_m$	Chance number of maintenance	$r_{m,n}$	Standard deviation-to-mean ratio for interval duration
$c_p$	Chance number of production	$k$	Classification coefficient
$c_s$	Chance number of setup	$k_{opt}$	Optimal classification coefficient
$c_t$	Chance number of stop	$n$	Number of open-closed cycles in a pattern
$c_w$	Chance number of waiting	$n_{opt}$	Optimal number of open-closed cycles in a pattern
$c_s$	Non-dimensionalized cumulative sum of intervals distribution	$\hat{n}$	Estimated number of open-closed cycles in a pattern
$d_{m,n}$	Interval duration mean	$nb_{open}$	Number of open intervals per bin
$d_{ts1,ts2}$	Difference between time series	$nb_{close}$	Number of close intervals per bin
$d_{ts1,ts2,crs}$	Cross difference between time series	$N$	Number of intervals
		$N_k$	Number of intervals classified as production time
		$N_{nprod,nrep}$	Number of non-production intervals correctly classified
		$N_{nrep}$	Total number of non-repetitive intervals
		$N_{prod}$	Total number of production intervals

✉ Emanuele Vaglio  
emanuele.vaglio@uniud.it

Marco Sortino  
marco.sortino@uniud.it

<sup>1</sup> Polytechnic Department of Engineering and Architecture, University of Udine, via delle Scienze 206, Udine 33100, Italy

$N_{prod,rep}$	Number of production intervals correctly classified
$OEE$	Overall Equipment Effectiveness
$OEE^*$	Overall Equipment Effectiveness neglecting quality factors
$p_{y_n}$	Sequence half-width
$p_{opt}$	Optimal $y_n$ sequence half-width
$p_m$	Probability of maintenance status
$RI$	Relative increment of $N_k$
$s_{max}$	Reference variance threshold for testing open-close patterns
$s_{ref,seq}$	Estimated standard deviation of combined interval durations
$s'_{ref,seq,p}$	Adjusted estimated standard deviation of neighborhood combined interval durations
$s_{x,close}$	Standard deviation of close interval durations
$s_{x,open}$	Standard deviation of open interval durations
$s_{y,n,p}$	Expected standard deviation of neighborhood combined interval durations
$s'_{y,n,p}$	Adjusted expected standard deviation of neighborhood combined interval durations
$s'_{y,n,opt,p}$	Adjusted expected standard deviation of neighborhood optimal combined interval durations
$s'_{y,p}$	Standard deviation of combined intervals
$s_{y,n}$	Expected standard deviation of combined interval durations
$x$	Interval durations
$y_n$	Combined interval duration
$y_{n,opt}$	Optimal combined interval duration
$\bar{y}_{y,n,p}$	Expected mean of neighborhood combined interval durations
$\mu_{BA}$	Average balanced accuracy
$\mu_{EBA}(i)$	Estimated average balanced accuracy
$\sigma_{BA}$	Standard deviation of the balanced accuracy
$\sigma_{EBA}$	Estimated standard deviation of the balanced accuracy
$DES$	Discrete Event Simulation
$IoT$	Internet of Things
$IIoT$	Industrial Internet of Things
$JSON$	JavaScript Object Notation format
$KPI$	Key Performance Indicator
$L - BFGS - B$	Limited-memory Broyden-Fletcher-Goldfarb-Shanno with Bound constraints
$MQTT$	Message Queuing Telemetry Transport protocol
$NoL$	Number of Levels
$ReLU$	Rectified Linear Unit

## 1 Introduction

Most of existing production plants still have limited connection capabilities and process engineers mainly rely on operators declarations, direct observations, and average throughput rates for driving improvement actions. The collection and processing of such data are rather demanding, it is affected by declaration inaccuracies, and the processing time may be unacceptably long. Moreover, production systems do evolve over time and this activity should be repeated from time to time to adapt to emerging bottlenecks.

Industry 4.0 and the smart manufacturing concept are strongly pushing for an interconnected system, but this mainly applies to new machines and installations. Legacy machines generally have very limited inter connectivity and data sharing capabilities. Moreover, the necessary investment for enabling those functions may be high. Therefore, there is a need for tools aimed at easily collecting data from existing plants and provide preliminary data for driving improvement actions and investments.

One possible solution may be to apply external sensors systems which do not require hardware or software modification of the machine. Very recently, Tran et al. proposed a comprehensive review of IoT-based approaches for condition monitoring [1]. Vibration sensors require minimal installation and are very promising for many applications, as reported by Er et al. in 2016 [2]. Unfortunately, vibration sensors are still rather expensive and require specific hardware for real-time processing of signals to extract relevant information.

Since energy consumption is becoming a critical aspect for environmental and economic reasons, many recent publications focus on the application of electrical power sensors for assessing and classifying the machines status in order to estimate their effective productivity and efficiency, as proposed by Abele et al. [3]. A systematic literature review of classification of machine statuses using energy consumption was recently proposed by Sihag et al. [4].

Teiwes et al. presented an application of machine learning for the identification of machines working statuses based on clustering the energy consumption data [5]. The cluster with the highest average power consumption was labeled as the “processing” state. The algorithm was based on many adaptive thresholds and confirmed the complexity of this application. In 2015, O’Driscoll et al. presented a non-intrusive approach for determining the operational status of a machine by measuring the main in-comer [6] and applying a statistical classifier. Sihag et al. [7] applied a similar approach in 2018 using unsupervised clustering. An unsupervised load monitoring approach for the classification of machine statuses was presented by Seevers et al. in 2019 [8].

A vertical grinding machine and a milling machine were used to collect data over one day and “in machining,” “ready,” and “warm-keeping” modes were detected. According to these works, the proposed strategies for energy consumption monitoring are very promising for increasing plant efficiency. Nevertheless, poor information regarding machines statuses classification accuracy was reported.

A more elaborate approach for the automatic assessment of machine tool energy efficiency and productivity was presented by Hacksteiner et al. in 2017 [9]. Internal machine data were collected in combination with data derived from additional external sensors. In detail, a power monitoring sensor was applied to the main incompressor while pressure and flow sensors were applied to the compressed air inlet of the machining center. Unfortunately, no performance evaluation of the automatic assessment system against other approaches was provided.

More recently, Petruschke et al. presented a machine learning approach for the classification of machine energy statuses, obtaining an accuracy higher than 95% [10]. This preliminary information also confirms the possibility of using energy monitoring systems to assess the efficiency of manufacturing plants in terms of productivity, in addition to estimating the absorbed electrical energy.

However, monitoring the energy consumption of a machine entails a substantial investment due to the costs of the sensing elements, ranging from hundreds to thousands of euros, the cost of the processing unit, and the cost of the electrical modifications of the machine. Moreover, the insertion of an electrical power measurement unit may be considered a major modification of the electrical plant; thus, it may imply a complete recertification and the loss of the warranty. For this reason, the electronics industry is proposing IIoT devices that can be easily mounted on production systems to start collecting readily available information such as the tower light status, the door status, the intensity of acoustic emissions, and several other metrics. A comprehensive review of sensor solutions for machine condition monitoring was recently published by Ahmad et al. [11]. All these systems are designed to be easy to install and interface with the factory network system to start recording information on a data repository. Nevertheless, how this data can be converted to relevant information is left to the user.

The main idea that started this research was that binary data obtained through simple sensors monitoring the machine tool door status may effortlessly provide a preliminary assessment of the machine performance and a rough evaluation of key performance indicators (KPIs) such as the overall equipment effectiveness (OEE) [12]. Indeed, the doors of machine tools are readily accessible and can be monitored with minimal sensorization investment while preserving the electrical certification of the equipment.

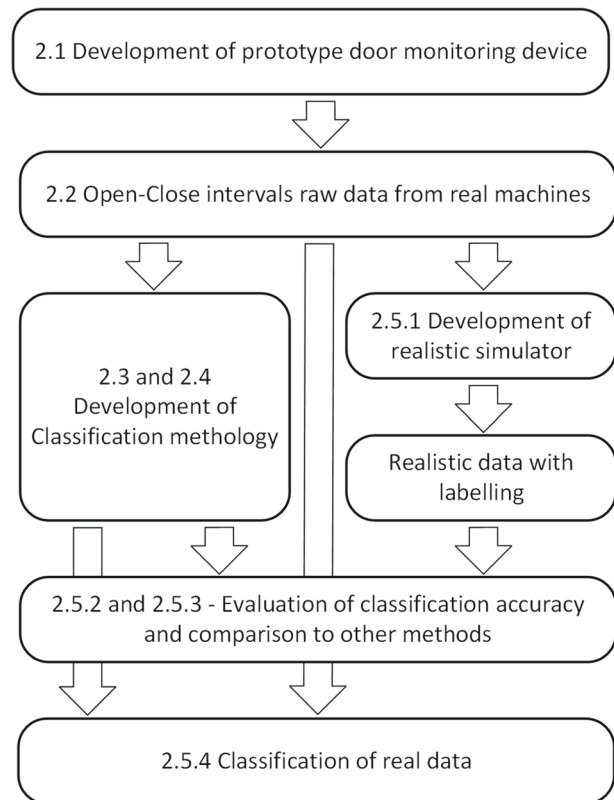
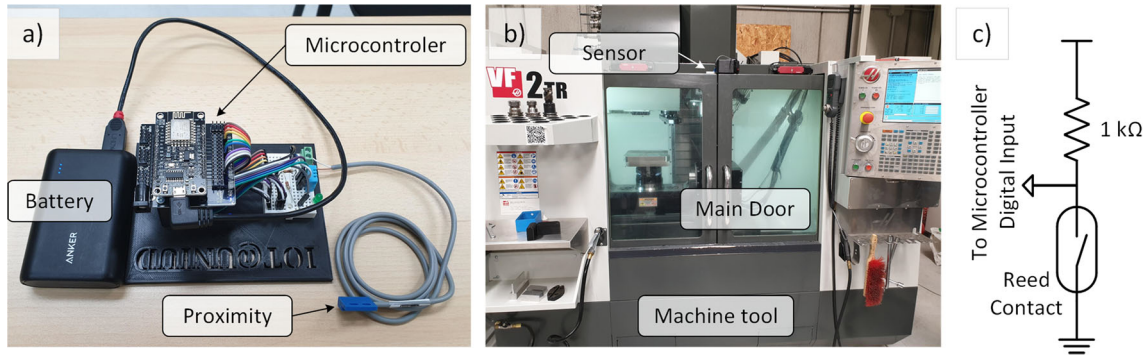


Fig. 1 Overview of the experimental approach

In the next sections, the development of a prototype data collection system designed to monitor the door status (open, close) is discussed. After laboratory validation, 50 devices were deployed to collect data from 50 machine tools in an operational automotive components factory for over 3 months. The collected data were analyzed and an innovative methodology for the automatic classification of machine status was proposed. A realistic simulator was developed to validate the classification methodology. Ultimately, the collected data were analyzed to calculate the efficiency of actual machines, with surprising and very promising results (Fig. 1).

## 2 Data collection and classification algorithm

In this section, the development of the monitoring device and its validation are discussed. Then, details about the application of sensors in an actual plant together with a preliminary data analysis are given. Finally, the development of the classification algorithm and its validation with using simulated data are provided.



**Fig. 2** Development of the monitoring device: (a) prototype test circuit; (b) installation of the device on the main door of a milling machine in laboratory; (c) electrical circuit of the reed contact sensor

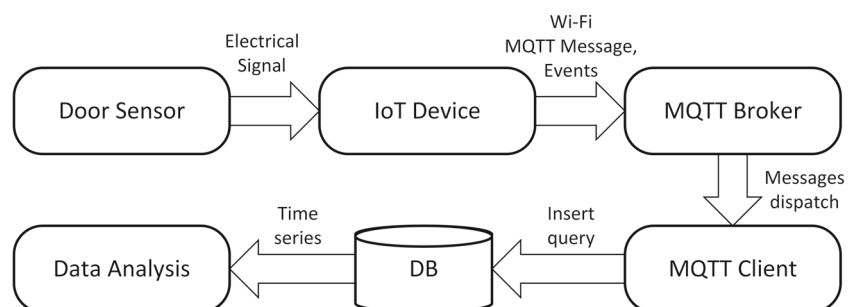
## 2.1 Door status monitoring devices

The status of the machine doors was monitored using simple and inexpensive magnetic reed switches. This type of sensor is priced at less than 30 USD and was selected due to the simplicity of installation – no precise mechanical fixtures or complex electrical conditioning circuits were necessary. The sensor was interfaced with an ESP8266 micro-controller which provided real-time data processing and Wi-Fi connectivity. The prototype system is shown in Fig. 2(a). It is noteworthy that the same simple device can be straightforwardly used for processing data from multiple other sensors.

In Fig. 2(b), the installation of the sensor on the door of a milling machine inside the laboratory is shown. The micro-controller and conditioning electronics were enclosed inside a plastic box for protection. Electrical power was provided either by a USB power supply or by a power pack.

The reed switch was connected to one of the digital inputs of an ESP8266 micro-controller with a pull-up resistor. The electrical circuit is shown in Fig. 1(c). The micro-controller was programmed to check the sensor status at approximately 1 kHz. If the sensor status varied from that in the previous scan cycle, an event message was sent to a message queuing telemetry transport (MQTT) broker in JavaScript Object Notation (JSON) format using Wi-Fi. To avoid bouncing, sensor state variations occurring within 200 ms from the last state change were ignored. The final cost for each prototype device amounted to less than 100 USD.

**Fig. 3** IIoT architecture and main components of the monitoring system



The JSON message included the unique micro-controller code, the sensor code, the sensor status, and the time interval in milliseconds from the last state change. The micro-controller programming was performed in C++ using the Arduino Integrated Development Environment.

The overall IIoT architecture of the system is shown in Fig. 3. On the server side, an MQTT client running on the same virtual machine of the broker received all JSON messages, decoded them, and inserted the new values into a PostgreSQL database table.

## 2.2 Application of sensors to a real plant

After preliminary testing and laboratory validation, 50 sensors were deployed onto the main doors of various types of machine tools within an operational factory specialized in manufacturing automotive components. The apparatus comprised injection molding machines, milling machines, machining centers, lathes, and grinding machines. Specific details about the equipment cannot be disclosed due to contractual confidentiality agreements. Manufacturing operations were organized in batches, with production changes occurring once or twice a month on some of the machines.

Plant production data were manually recorded on paper, with daily productivity and scrap rate information being partially transferred to Excel spreadsheets on a monthly basis. The majority of the machine tools were outdated, and the anticipated investment required for interconnecting even the



newer machines was significant. Consequently, this monitoring activity also aimed at providing a preliminary efficiency assessment of the plant. Specifically, an estimation of losses resulting from material shortages, cleaning procedures, and minor maintenance tasks was expected.

Some of the doors of these machines were automated and door opening and closing was automatically controlled. In many machines, the closing of the door directly initiated or resumed the execution of the part program. Additionally, the open-close logic varied among machines due to their distinct characteristics and configurations, as it relied on operational or installation details.

The 50 machines were monitored continuously for over 3 months, and 1 GB of data was collected. No problems with connectivity or power loss were registered in this time interval.

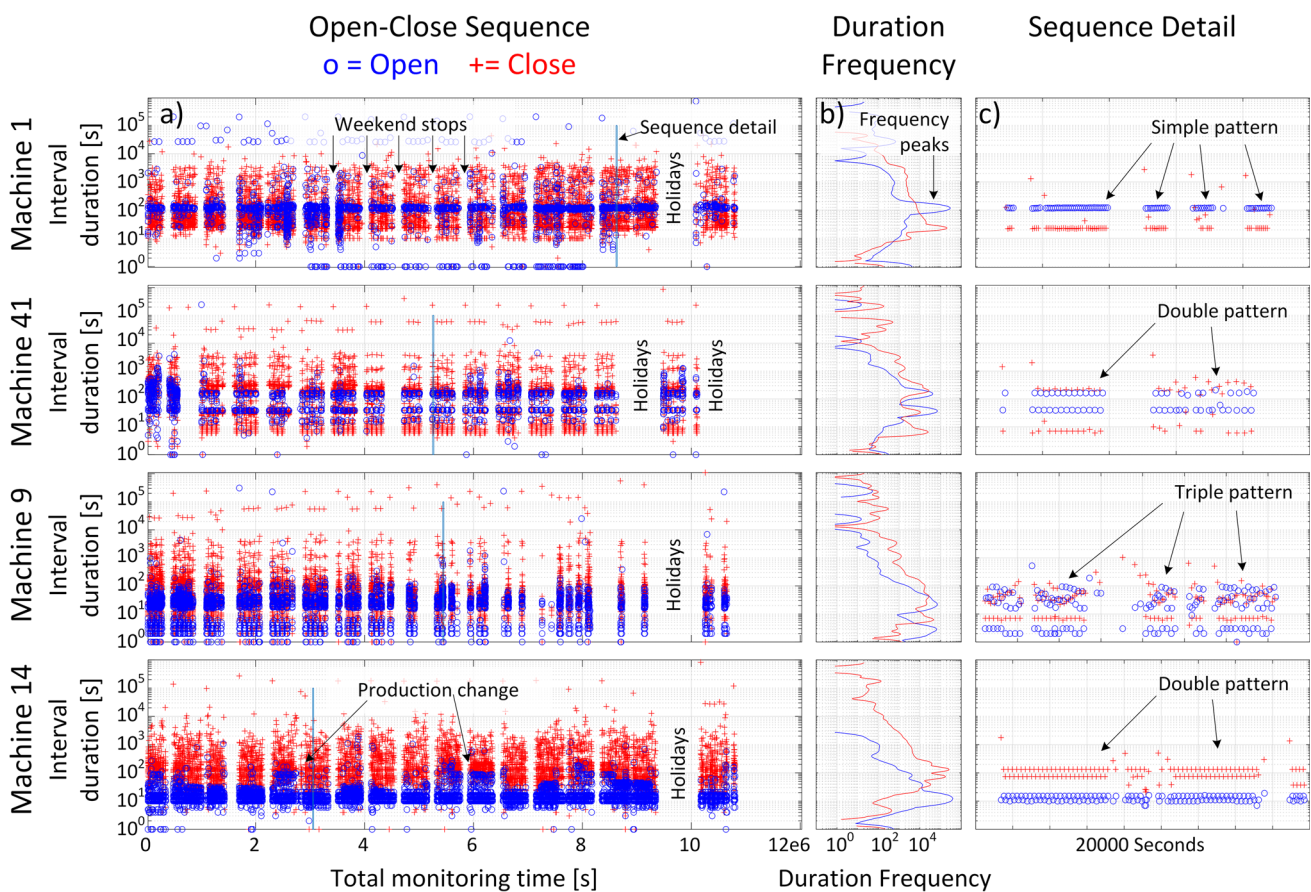
Data were exported from PostgreSQL in CSV format and imported into the Mathworks MATLAB environment, where all analysis activities were performed. As a preliminary stage, the data were filtered and processed to produce a final dataset

of approximately 80 MB, detailing solely the sequence of open-close intervals for each machine. This dataset was structured as a table with three columns: the Unix timestamp of the interval end, the interval type (0=open door; 1=closed door), and the interval duration in seconds.

### 2.3 Preliminary analysis of the sensor data

Data analysis was performed to highlight repetitive time patterns and to identify methodologies for the automatic classification of machine statuses. The data collected from different machines varied strongly, and some examples are illustrated in Fig. 4, where a logarithmic scale was employed to effectively represent a broad range of time intervals, spanning from seconds to several days.

In Fig. 4, the open-close interval duration data are represented with different colors, using the y-axis to represent the interval duration in seconds, and the x-axis to provide the position of the interval end relative to the monitoring time. In this way, each interval corresponds to a single point



**Fig. 4** Examples of data collected from machines 1, 41, 9, and 14: (a) open-close intervals duration versus monitoring time. The x-coordinate represents the interval end position. Sequences of intervals with the same duration appear as adjacent points on a horizontal line, facilitating the identification of repetitive patterns; (b) frequency analysis

of the interval duration enabling the identification of typical duration peaks associated with repetitive patterns; (c) detail of the open-close sequence obtained focusing on a time range of 20,000 s to illustrate the repetitive pattern. The y-axes of charts (b) and (c) are aligned with that of chart (a)

on a graph and sequences of open-close intervals are represented by point patterns. A sequence of intervals with the same duration is represented as a set of neighboring points on the same horizontal line, making it is easier to identify repetitive patterns. For this reason, duration frequency analysis was performed to identify the duration peaks typical of repetitive patterns.

The following observations were made:

1. For all machines, long intervals corresponding to production stops, holidays, weekends, and night shifts were identified. These intervals ranged from hours to several days.
2. The time schedule of the machines were quite different with different shifts (2–3 shifts per day). Some systems were active during the weekend.
3. Almost all machines evidenced repetitive patterns. These patterns represent normal repetitive production operations during batch production, and they were separated by long time intervals without movement or by sequences of non-repetitive time intervals that were explained as setups, cleaning, and maintenance operations.
4. In several machines, production changes were clearly discernible, as different repetitive patterns separated by chaotic patterns were visible.
5. As illustrated in Fig. 5, the repetitive patterns were of different kinds: simple open-close cycles, where both the open and close intervals had quasi-constant durations; double open-close-open-close cycles, where the open and close intervals had two very distinct alternating durations; triple open-close-open-close-open-close cycles with an approximately constant overall duration; semi-repetitive cycles, where either the open or close interval duration was approximately constant while the other interval duration was widely distributed; and chaotic cycles, where the duration frequency distributions of both intervals were rather large.

Due to the presence of repetitive patterns, frequency analysis of the time intervals was applied to identify peaks in the distribution diagram. For instance, in Fig. 4, machines with

different logic are shown. Machine 1 exhibited a repetitive pattern of single open-close sequences, with the open time interval demonstrating remarkable regularity, as indicated by the narrow frequency peaks. Machine 14 was a perfect example of double open-close pattern and two very narrow frequency peaks for both the open and close intervals are visible. Machine 9 had a complex triple open-close pattern and a rather complex overall behavior with many non-production intervals. Machine 41 displayed a notably precise double open-close pattern and discernible production changes.

These four cases were selected to provide the reader with an idea of the data quality and variability and lay the groundwork for explaining the requirements of the automatic classification algorithm described in the next section.

## 2.4 Automatic classification algorithm

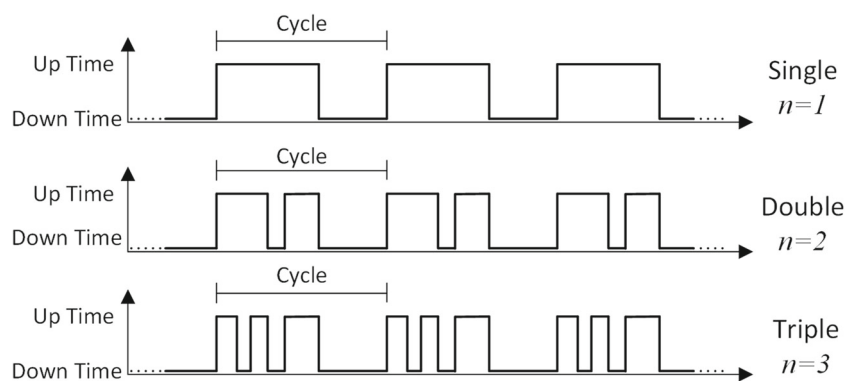
After the preliminary data analysis, the following requirements for the automatic classification algorithm were identified:

- The algorithm should adapt to different open-close repetitive patterns and production changes, but it should not rely on the type of intervals, since many machines exhibit inverse or varying logic.
- For the greatest applicability of the methodology, it should be possibly based on adaptive and non-dimensional thresholds.

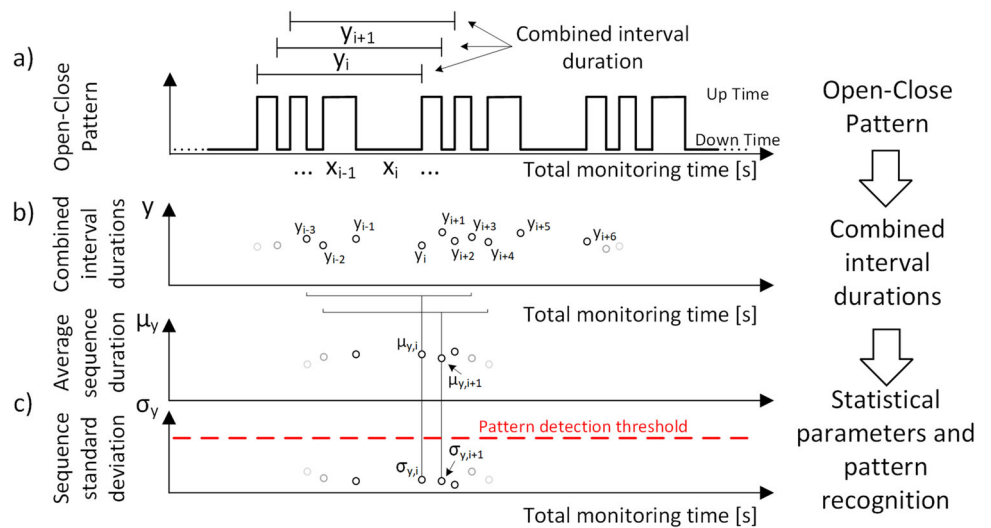
To simplify the approach, it was assumed that the logic of each machine was constant in the monitoring interval. This was not considered a strong limitation since it could be easily overcome by splitting production intervals into smaller time frames, if necessary.

Among all the possible classification approaches, a statistical approach based on the intervals distributions and joint distributions was attempted. Many alternative ways were attempted to establish a statistical classification criterion, and eventually the procedure described in Figs. 6 and 7 was developed, where  $x_i$  with  $i = 1 \dots N$  is the sequence of interval durations recorded from one machine.

**Fig. 5** Description of the different open-close repetitive patterns



**Fig. 6** Description of open-close data used to determine statistical parameters and identify patterns. Open-close patterns are first transformed into combined interval duration sequences, then the average and the standard deviation are computed to enable the identification of patterns. The detection threshold determines which open-close cycle patterns ( $n$ ) provide the best classification performance



As shown at the top of Fig. 6, if a repetitive pattern composed of  $n$  open-close cycles is present, the average overall duration should be rather constant. Since the data evidenced  $n = 1, 2, 3$  open-close cycle patterns, all these possibilities had to be tested to determine which pattern provided the best classification performance, i.e., which one had a variance of neighboring points lower than a reference threshold  $s_{max}$ .

The detailed step-by-step analysis process is shown in Fig. 7. The first aim of this process is to provide a suitable starting estimate for  $s_{ref,seq}$ . For this purpose, the long intervals are first identified. The following long interval classification was adopted:

- Long stops: duration between 2 and 6 h;
- Missing shifts: duration between 6 and 10 h;
- Missing double shifts: duration between 10 and 20 h;
- Free days: duration between 20 and 32 h;
- Weekends or 2-day breaks: duration between 32 and 56 h;
- Holidays: duration over 56 h.

Unfortunately, precise data were not available in this case, and the schedule varied widely from machine to machine. However, it is highly advised that long interval classification is performed in adherence to production schedule.

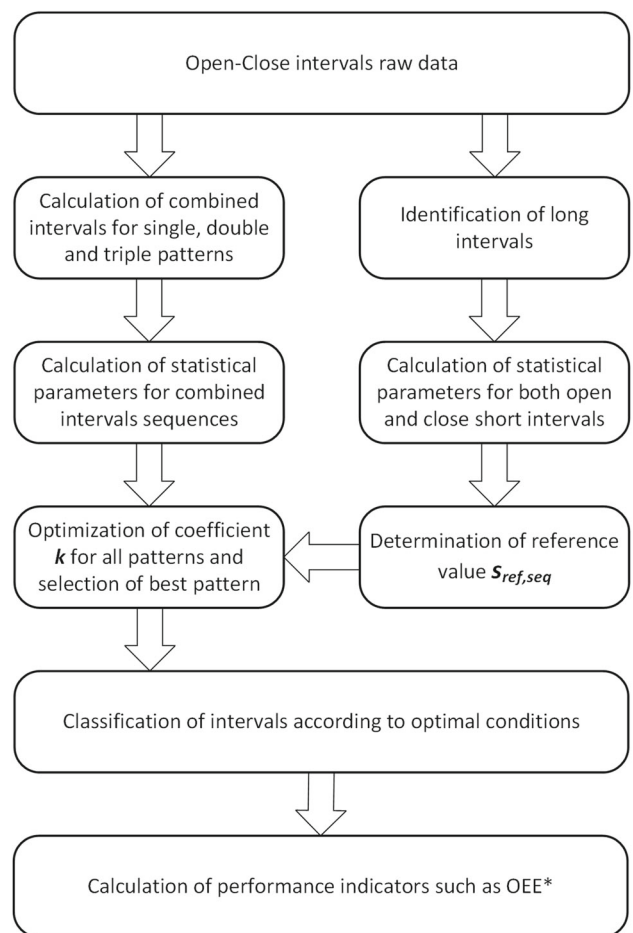
After classification of the long intervals, the method focused on short open and close intervals and the standard deviations of the open and close intervals,  $s_{x,open}$  and  $s_{x,close}$ , were calculated.

Let the combined interval duration  $y_{i,n}$  be defined as follows:

$$y_{i,n} = \frac{\sum_{i-2n+1}^i x_i}{2n} \text{ and } n = 1, 2, 3 \tag{1}$$

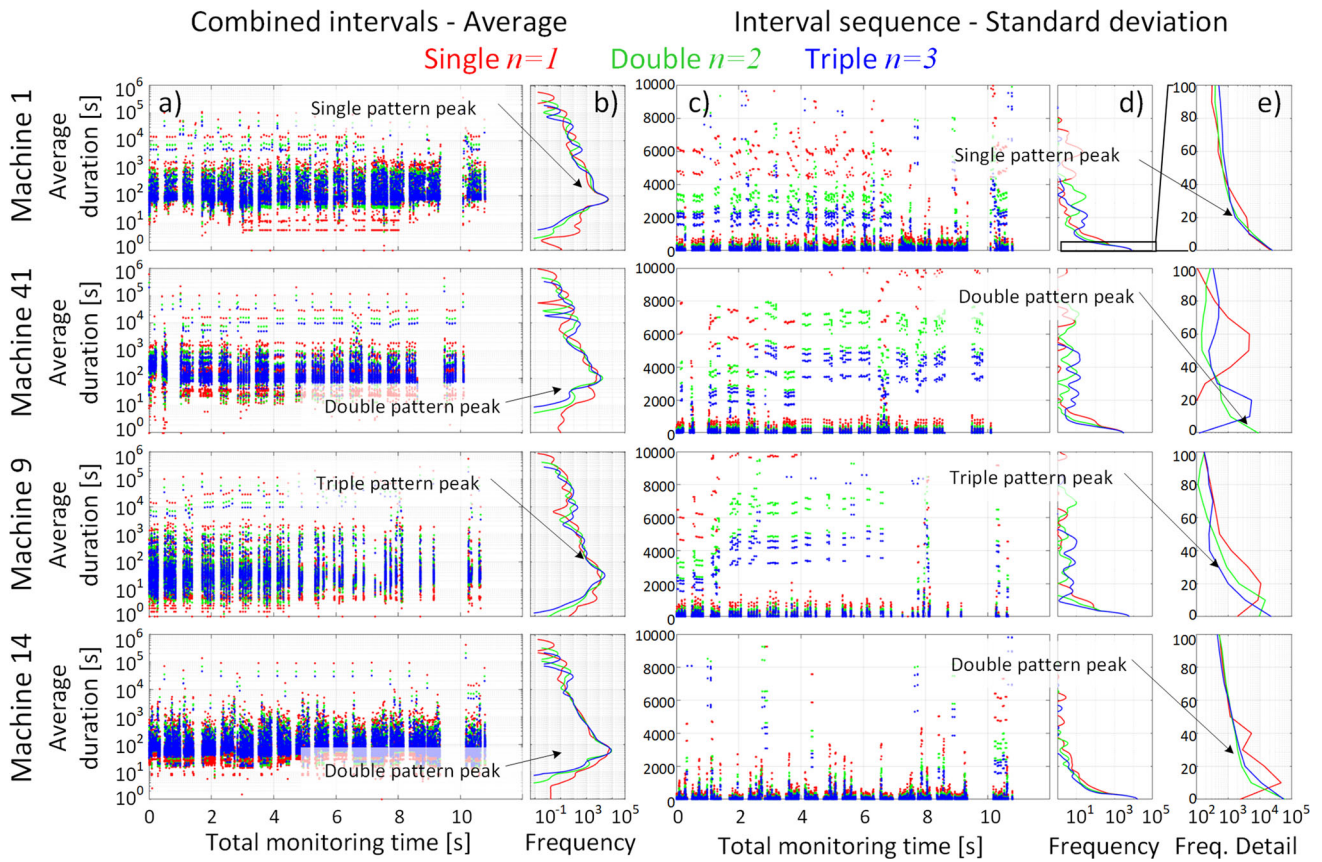
The characteristics of the combined interval duration are given in Fig. 8(a) and (b) using a logarithmic scale to represent a wide spectrum of time intervals. In Fig. 8(a), the

combined interval durations are shown for  $n = 1, 2, 3$ . The dispersion is lower for  $n = 3$  due to the smoothing effect of averaging. In Fig. 8(b), the frequency distribution of the



**Fig. 7** Analysis of open-close data used to determine statistical parameters, identify patterns, classify intervals, and calculate performance indicators





**Fig. 8** Example of combined intervals durations for machines 1, 41, 9, and 14: (a) combined intervals with different open-close cycle patterns ( $n$ ) of lengths  $y_{i,n}$ ; (b) frequency analysis of the combined intervals duration showing very distinctive duration peaks associated with repetitive patterns; (c) standard deviations of sequences of combined intervals  $s'_{y,i,p}$  versus the total monitoring time taking consistently low

values during repetitive production; (d) frequency analysis of standard deviations in (c) focusing on the occurrence of a peak; (e) detail of the frequency analysis in (d) showing the open-close cycle patterns detected. The y-axis of charts (b) is aligned with that of chart (a), whereas the y-axis of chart (d) is aligned with that of charts (c)

interval durations is shown. For all the example machines, there were very distinctive peaks typical of repetitive patterns.

The expected standard deviation  $s_{y,n}$  is as follows:

$$s_{y,n} = \frac{1}{2\sqrt{n}} \sqrt{s_{x,open}^2 + s_{x,close}^2} \quad (2)$$

Let us now consider a sequence of  $y_{i,n}$  values composed of the  $p$  preceding values and the  $p$  subsequent values for each element  $i$ , for a total of  $2p + 1$  elements. The statistical parameters of the sequence are:

$$\bar{y}_{y,i,n,p} = \frac{\sum_{i-p}^{i+p} y_{i,n}}{2p + 1} \quad (3)$$

and

$$s_{y,i,n,p} = \sqrt{\frac{\sum_{i-p}^{i+p} (y_{i,n} - \bar{y}_{y,i,n,p})^2}{2p + 1}} \quad (4)$$

Since the  $y_{i,n}$  elements in the expression of  $s_{y,i,n,p}$  are not mutually independent, it is rather complex to determine the expected value of  $s_{y,i,n,p}$  [13]. Preliminary investigations evidenced that it depends linearly on  $\frac{1}{n}$ ; therefore, the sequence standard deviation was adjusted to have a reference that does not depend on  $n$ , as follows:

$$s'_{y,i,n,p} = n \cdot s_{y,i,n,p} \quad (5)$$

The characteristic of the standard deviation for each machine introduced in Fig. 4 is given in Fig. 8(c), (d), and



(e). In Fig. 8(c), the standard deviation against the total monitoring time is provided. During repetitive production, the values were consistently low, and a peak was observed in the frequency distribution, as illustrated in Fig. 8(d) and further detailed in Fig. 8(e).

To compare the value of  $s'_{y,i,n,p}$  with a reference value, Eq. 2 was modified to remove  $n$  and to take into account the effect of averaging over  $2p + 1$  values. The adjusted expected standard deviation of the sequence  $s'_{ref,seq,p}$  was estimated as follows:

$$s'_{ref,seq,p} = \frac{1}{2\sqrt{2p+1}} \sqrt{s_{x,open}^2 + s_{x,close}^2} \tag{6}$$

Unfortunately, this reference value demonstrated some degree of unreliability owing to inaccuracies inherent in the estimates of  $s_{x,open}$  and  $s_{x,close}$ . Therefore, the coefficient  $k$  was introduced to allow an adaptable classification approach. Accordingly, a sequence centered on the  $i$ -th element was considered repetitive if:

$$s'_{y,i,n,p} \leq k \cdot s'_{ref,seq,p} \tag{7}$$

All the raw intervals composing a sequence which satisfies (7) can be classified as “production time.” Let  $N_k$  represent the number of intervals falling under this classification.

In Fig. 9(a), the fraction of short intervals classified as production time for increasing values of  $k$  are given for the four reference machines. For each  $n$ , the value of  $k$  was increased

from 0.01 to 1.5 in steps of  $\Delta k = 0.01$ . As expected, the fraction of intervals classified as “production time” increased with increasing values of  $k$ .

The pattern characteristics were significantly different across different apparatus. For machines 41, 9, and 14, the characteristic corresponding to the effective pattern demonstrated superior classification performance for small values of  $k$ . Conversely, for machine 1, the three characteristics were nearly superimposed.

Figure 9(b) shows the relative increment  $RI$  calculated for each value of  $k$ :

$$RI(k) = \frac{N_k - N_{k-\Delta k}}{N_{k-\Delta k}} \tag{8}$$

All the characteristics of  $RI$  generally decrease with increasing values of  $k$ . In the left part of each diagram, when  $k$  is close to 0, the characteristics have a clear and strong decreasing trend, especially those that correspond to the pattern type of the machine. After this decreasing trend, all characteristics slightly decrease asymptotically with a very noisy behavior. The transition between the initial decreasing trend and the noisy behavior was found to be a good marker separating the intervals effectively belonging to a repetitive pattern from the others. Accordingly, the non-dimensional threshold of  $RI \leq 0.01$  was adopted to determine the optimal value of  $k$ .

For machines 41, 9, and 14, the curve corresponding to the pattern of the machine also had a  $k_{opt,n}$  that was significantly

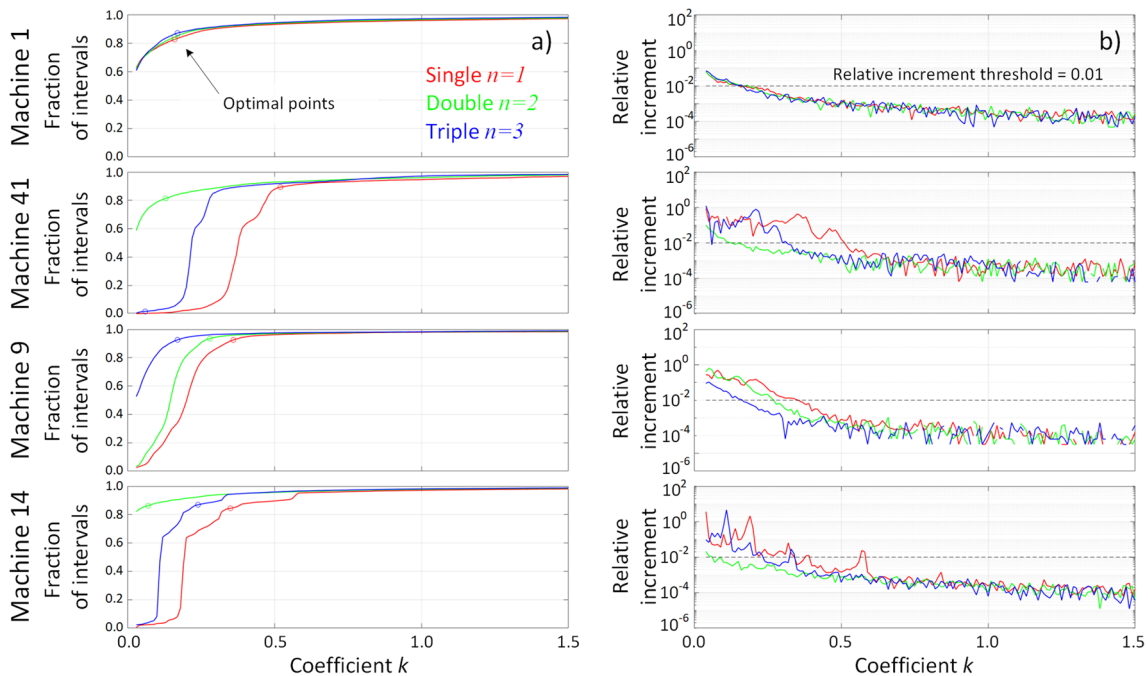


Fig. 9 Description of open-close data used to determine statistical parameters and identify patterns

smaller than the others, thus further confirming the validity of the proposed approach. For machine 1, the three  $k_{opt,n}$  values were almost identical. The conclusion was that for the pattern  $n = 1$  all curves provided a very similar classification which is a consequence of a single cycle pattern. In other cases, the curves that first reached saturation for a relatively small value of  $k$  corresponded to the most likely type of pattern, which was then assigned to that machine for that time interval.

It should be pointed out that the values of  $k$  and  $s'_{ref,seq,p}$  are determined dynamically for each machine and they are adaptive non-dimensional thresholds, as previously specified.

In conclusion, this procedure was used to determine the corresponding pattern from the raw dataset of each machine, and to provide a suitable classification of short intervals. At the end of this process, the short intervals were classified as belonging to a repetitive pattern, either marked as production time or not.

### 3 Results and evaluation of the algorithm performance

Unfortunately, no real data characterizing the effective status of each machine tool during the production process were available. Therefore, for evaluating the classification accuracy of the novel algorithm, a numerical simulator aimed at generating realistic synthetic data was developed. In this way, it was possible to quantitatively assess its classification capabilities. It is worth noting that only the data generated by the simulator that were similar to real data were kept for the final performance evaluation, as illustrated in the following section.

### 3.1 Development of the simulator

The simulator was completely developed in MATLAB working environment using a discrete event simulation (DES) approach. Starting from an initial idle condition, the simulator determined the new status of the machine from the possible alternatives according to their respective probabilities. Door movements were then determined based to machine status. The logical statuses of the machines and other simulation details are described in Table 1.

At each simulation step, a random number was generated to determine the next status of the machine according to its respective frequency number. For instance, the effective probability of maintenance  $p_m$  is:

$$p_m = \frac{c_m}{c_m + c_p + c_s + c_t + c_w} \quad (9)$$

where  $c_m$ ,  $c_p$ ,  $c_s$ ,  $c_t$ , and  $c_w$  are the random frequency numbers of maintenance, production, setup, stop, and waiting, respectively.

Each simulation produced synthetic data covering a time interval of approximately 4 months including shifts and weekend breaks.

The simulator was run 10,000,000 times with different combinations of the simulation parameters using Hammersley sampling [14, 15] and according to the overall design of the experiments in Table 2.

For each simulation, the output of the simulator was a long table (approximately 30,000 rows on average) where each row was an interval and the columns were the start time stamp of simulated interval, the door status, the interval duration, and the simulated machine status.

**Table 1** List of machine statuses implemented in the simulation: status name; description; number of simulated intervals to be simulated; simulated interval duration range; interval duration distribution

Status	Description	Int. Num.	Int. Dur. [s]	Distrib.
Idle	Machine status at the beginning of the simulation run and during non-scheduled time intervals;	N/A	N/A	N/A
Setup	Random activities during machine preparation for production;	< 12	100–300	Uniform
Production	Repetitive activities according to the simulated open-close pattern;	2, 4, 6	Simul. params.	Normal
Stop	Short production interruption for minor cleaning or other ordinary activities;	1	1–10	Uniform
Wait	Production interruption due to absence of incoming material from previous production stages;	1	10–1,800	Uniform
Maintenance	Production interruption due to breakdown and sequence of random activities during repair.	8	1,800–7,200	Uniform

For production status, the reference interval duration range was derived from factors in Table 2

**Table 2** Experimental design for simulation: factors and symbols; levels for each factor; number of levels (NoL)

Factor and symbol	Levels	NoL
Pattern number $n$	1, 2, 3	3
Chance number of maintenance $c_m$	0.00005, 0.0001, 0.0002, 0.0003, 0.0005, 0.001, 0.002, 0.003, 0.005, 0.01	10
Chance number of production $c_p$	4	1
Chance number of setup $c_s$	0.0002, 0.0005, 0.001, 0.002, 0.003, 0.005, 0.01, 0.02	8
Chance number of stop $c_t$	0.0005, 0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.08	8
Chance number of waiting $c_w$	0.0005, 0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.08	8
Interval duration mean $d_m, n$	2, 3, 5, 8, 10, 15, 20, 30, 50, 80, 100, 120, 150, 200, 250, 300, 400, 500, 600	19
Interval duration st.dev./mean ratio $r_m, n$	0.1, 0.2, 0.3, 0.4, 0.5, 0.75, 1.0, 1.25, 1.5, 2, 2.5, 3, 4, 5	14

The data structure was identical to that acquired from real machines, with the addition of a machine status column that was used to evaluate the performance of the classification algorithm.

In total, the simulations lasted approximately 30h on a PC equipped with an 8-core microprocessor running at 3.8-GHz clock speed. The simulations generated around 10 TB of data, making it impractical to store all data on disk. Consequently, only simulations with data resembling real-world scenarios were recorded and used for subsequent analysis. For this purpose, a similarity criterion was introduced to evaluate the affinity between time series derived from measurement and simulation. The criterion was based on the difference between the non-dimensional cumulated distributions of the interval durations of the two time series under analysis. To apply this criterion, the following steps were necessary:

- A set of classification bins was determined according to the characteristics of the available data. The bins ranged from 1 to 7,200s to cover short intervals. This range was divided into 1,000 intervals on a logarithmic scale. Since fractional numbers were useless in this case, each point was rounded to the closest integer, and duplicates were removed. Ultimately, 580 bins were obtained.
- The numerosity of the intervals belonging to each bin was determined. This operation was performed for open and close intervals separately, thus providing two arrays of 580 elements representing the original time series— $nb_{open}(i)$  and  $nb_{close}(i)$ . Each array was non-dimensionalized by dividing it by the respective total number of intervals. Ultimately, the cumulative sum distribution of each array was calculated:

$$cs_z(i) = \frac{\sum_{j=1}^i nb_z(j)}{\sum_{j=1}^{580} nb_z(j)}, z \in (open, close) \tag{10}$$

- The difference between two cumulative sum arrays, either referring to open or close intervals, was calculated

as follows:

$$d_{z,w} = \sum_{i=1}^{580} |cs_z(i) - cs_w(i)| \tag{11}$$

where the  $z$  and  $w$  symbols generally indicate two distinct cumulative distributions.

- Both the straight and cross differences were calculated. The straight difference was obtained by the composition of corresponding difference terms:

$$d_{ts1,ts2,str} = \sqrt{d_{ts1=o,ts2=o}^2 + d_{ts1=c,ts2=c}^2} \tag{12}$$

The cross difference was obtained by the composition of the cross terms:

$$d_{ts1,ts2,crs} = \sqrt{d_{ts1=o,ts2=c}^2 + d_{ts1=c,ts2=o}^2} \tag{13}$$

- The difference between two time series was eventually obtained as the minimum between straight and cross differences:

$$d_{ts1,ts2} = \min(d_{ts1,ts2,str}, d_{ts1,ts2,crs}) \tag{14}$$

Simulations were only recorded if their distance from at least one of the real time series was below 30. This threshold was determined by analyzing the average distances between real time series. As a result, a total of 18,695 simulations were acquired, comprising 18.6 GB of data in memory or 2.4 GB on disk when stored in compressed format.

### 3.2 Algorithm performance on synthetic data

The classification accuracy of the algorithm was calculated by using the subset of synthetic data that was close to real data. Accuracy was defined as the ability to correctly classify production intervals as “repetitive” and all other intervals as

“non-repetitive.” The balanced accuracy BA was then computed as follows:

$$BA = 0.5 \left( \frac{N_{prod,rep}}{N_{prod}} + \frac{N_{nprod,nrep}}{N_{nrep}} \right) \tag{15}$$

where  $N_{prod,rep}$  is the number of production intervals correctly classified as “repetitive”;  $N_{prod}$  is the total number of production intervals;  $N_{nprod,nrep}$  is the number of non-production intervals correctly classified as “non-repetitive”;  $N_{nrep}$  is the total number of “non-repetitive” intervals. The balanced accuracy index ranges from 0.5 (worst) to 1 (optimal).

For each synthetic time series, the procedure described in Fig. 7 was applied and the following quantities were obtained:

- The estimated pattern number  $\hat{n}$ ;
- The best sequence half-width  $p_{opt}$  and coefficient  $k_{opt}$ ;
- The balanced accuracy BA.

In Table 3, the classification performance data are given against the true and estimated pattern numbers.

Let us first consider the elements on the diagonal: in 4,141+2,647+231=7,019 cases out of 18,695 the pattern type was correctly recognized (37.54%). For  $n = 1$ , the majority of the simulated time series were classified incorrectly. However, as discussed in the previous section, this was not a problem since the single pattern classification performance was similar for all  $n$ .

The elements outside the matrix diagonal for  $n = 2$  and  $n = 3$  cannot be considered acceptable, totaling 137+501+6+31=675 cases out of 18,695 (3.6%). The failure to recognize the correct pattern could be attributed to large standard deviations or very similar average interval durations among different elements within the same cycle. In conclusion, the pattern was correctly identified in 96.4% of cases.

Additionally, the average balanced accuracy  $\mu_{BA}$  and the standard deviation  $\sigma_{BA}$  were computed according to conventional definitions. In all cases,  $\mu_{BA}$  exceeded 90%, while  $\sigma_{BA}$  was very low. This finding demonstrates that, on average, the classification algorithm performed very well, even in complex scenarios where the underlying pattern was unclear.

The classification algorithm was executed using different levels (1, 2, 3, and 4) of the half-width sequence  $p$  to determine the optimal value. The results shown in Fig. 10 indicate that the cumulated distribution improves with increasing values of  $p$ . However, a slight decrease in performance was observed when comparing  $p = 4$  to  $p = 3$ . Consequently,  $p = 3$  was adopted as optimal value.

Concerning the coefficient  $k$ , since it was automatically adapted for each classification, its values are not particularly interesting. However, for a rough reference, the average value was 0.17, with a maximum of 0.75 observed among all considered cases.

### 3.3 Preliminary comparison of algorithm performance against other classification approaches

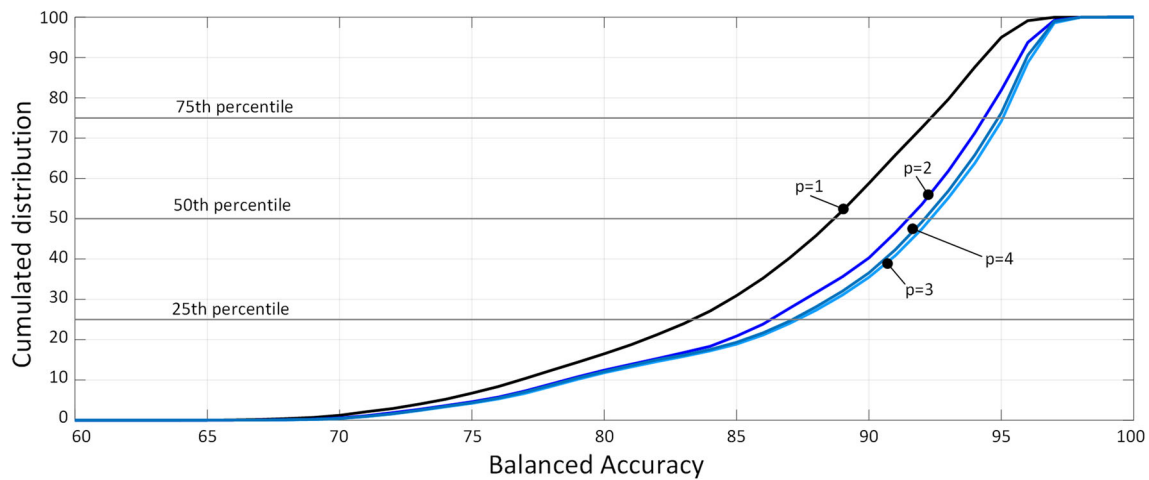
To provide a more comprehensive evaluation of the algorithm performance, a preliminary comparison of its accuracy against that of other classifiers was performed.

For each realistic simulation, the  $y_{i,n_{opt}}$  and  $s'_{y,i,n_{opt},p}$  and the simulated machine status values were used to train a random forest of 10 classification trees using cross-validation in Matlab environment. The data corresponding to the optimal number  $n = n_{opt}$  were used. Training and evaluation were performed separately for each simulation, thus maximizing the classification capabilities on homogenous data. The balanced accuracy of the classification performed by the random forest against that obtained with the proposed methodology is given in Fig. 11.

**Table 3** Classification performance data against true pattern number and estimated pattern number

		Estimated pattern			Total
		$\hat{n} = 1$	$\hat{n} = 2$	$\hat{n} = 3$	
True pattern	$n = 1$	$C = 4, 141$	$C = 4, 798$	$C = 6, 203$	$C = 15, 142$
		$\mu_{BA} = 0.906$	$\mu_{BA} = 0.923$	$\mu_{BA} = 0.917$	$\mu_{BA} = 0.916$
		$\sigma_{BA} = 0.064$	$\sigma_{BA} = 0.059$	$\sigma_{BA} = 0.062$	$\sigma_{BA} = 0.062$
	$n = 2$	$C = 137$	$C = 2, 647$	$C = 501$	$C = 3, 285$
		$\mu_{BA} = 0.915$	$\mu_{BA} = 0.910$	$\mu_{BA} = 0.925$	$\mu_{BA} = 0.913$
		$\sigma_{BA} = 0.065$	$\sigma_{BA} = 0.063$	$\sigma_{BA} = 0.051$	$\sigma_{BA} = 0.062$
$n = 3$	$C = 6$	$C = 31$	$C = 231$	$C = 268$	
	$\mu_{BA} = 0.909$	$\mu_{BA} = 0.915$	$\mu_{BA} = 0.918$	$\mu_{BA} = 0.918$	
	$\sigma_{BA} = 0.045$	$\sigma_{BA} = 0.055$	$\sigma_{BA} = 0.056$	$\sigma_{BA} = 0.055$	





**Fig. 10** Cumulated distributions of balanced accuracy obtained with different half-width sequence  $p$

The performance of the two methodologies is almost equivalent, with the random forest performing better in cases where the proposed methodology is not very accurate, probably due to the specialization of the classification tree. Overall, the effectiveness of the proposed methodology is, on average, at least equivalent to that of a random forest approach.

In order to compare the performance of the two methods on heterogeneous data, 100 intervals (50 production, 50 other) were selected randomly from each of the 18,695 realistic simulation runs, and combined together, thus obtaining a dataset of  $y_{i,n_{opt}}$  and  $s'_{y,i,n_{opt},p}$  and the simulated machine status values with 1,869,500 records. Again, a random forest of 10 classification trees using cross-validation was trained on this dataset in Matlab environment. The balanced accu-

racy of the random forest was 88.8% whereas that of the proposed methodology was 91.0%.

For further reference, an ensemble of 10 neural networks (2 inputs, 10 neurons in the hidden layer, ReLu activation function, L-BFGS-B solver, training set size 75%) was trained on a portion of the dataset, obtaining a balanced accuracy of 89.9%. The confusion matrices are reported in Table 4.

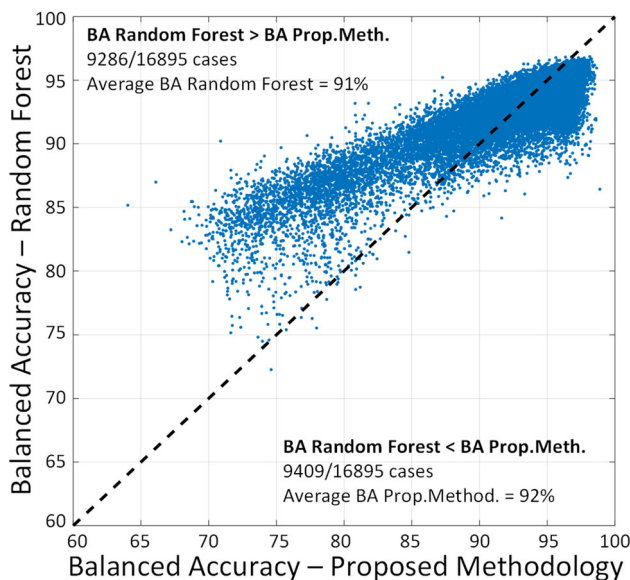
In conclusion, with homogeneous data, the performance of the proposed classification methodology is equivalent or slightly better than a random forest approach. With heterogeneous data, the performance of the proposed classification methodology evidences a higher balanced accuracy.

### 3.4 Estimated algorithm performance on real time series

Ultimately, the classification algorithm was employed on the real time series to classify the intervals and calculate their performance indicators.

In Fig. 12, relevant quantities describing the 50 real time series are shown. In particular, Fig. 12(a), (b), and (c) illustrate characteristic metrics offering insight into the dataset's variability and serving as a reference for subsequent diagrams. Here, a logarithmic scale was employed to effectively represent a broad range of time intervals, spanning from seconds to several days, and a diverse numbers of intervals, ranging from thousands to hundreds of thousands.

As the overall sampling interval is similar for all real time series, there is an obvious inverse correlation between the mean interval duration and the number of intervals. In addition, data derived from different machines appear to be rather heterogeneous, as clearly visible in the example data of Fig. 4. In Fig. 12(d), an estimate of the classification accuracy is shown. The estimation of the accuracy for the real



**Fig. 11** Comparison of balanced accuracy (BA) for classification of synthetic data

**Table 4** Classification performance data of proposed methodology against machine learning approaches

Approach		True classes		Balanced Accuracy
		True negative	True positive	
Proposed approach	Negative	794,050	27,271	$BA = 91.0\%$
	Positive	140,700	907,479	
Random forest	Negative	813,001	89,675	$BA = 88.8\%$
	Positive	119,462	847,362	
Neural network	Negative	796,353	52,808	$BA = 89.9\%$
	Positive	136,110	884,229	

time series was obtained by employing the weighted average method based on the accuracies of synthetic time series. The reciprocal of the difference between the synthetic and real time series was used as weighting factor:

$$\mu_{EBA}(i) = \frac{\sum_{j=1}^{18,695} \frac{BA(j)}{d_{tss=j,tsr=i}}}{\sum_{j=1}^{18,695} \frac{1}{d_{tss=j,tsr=i}}}$$

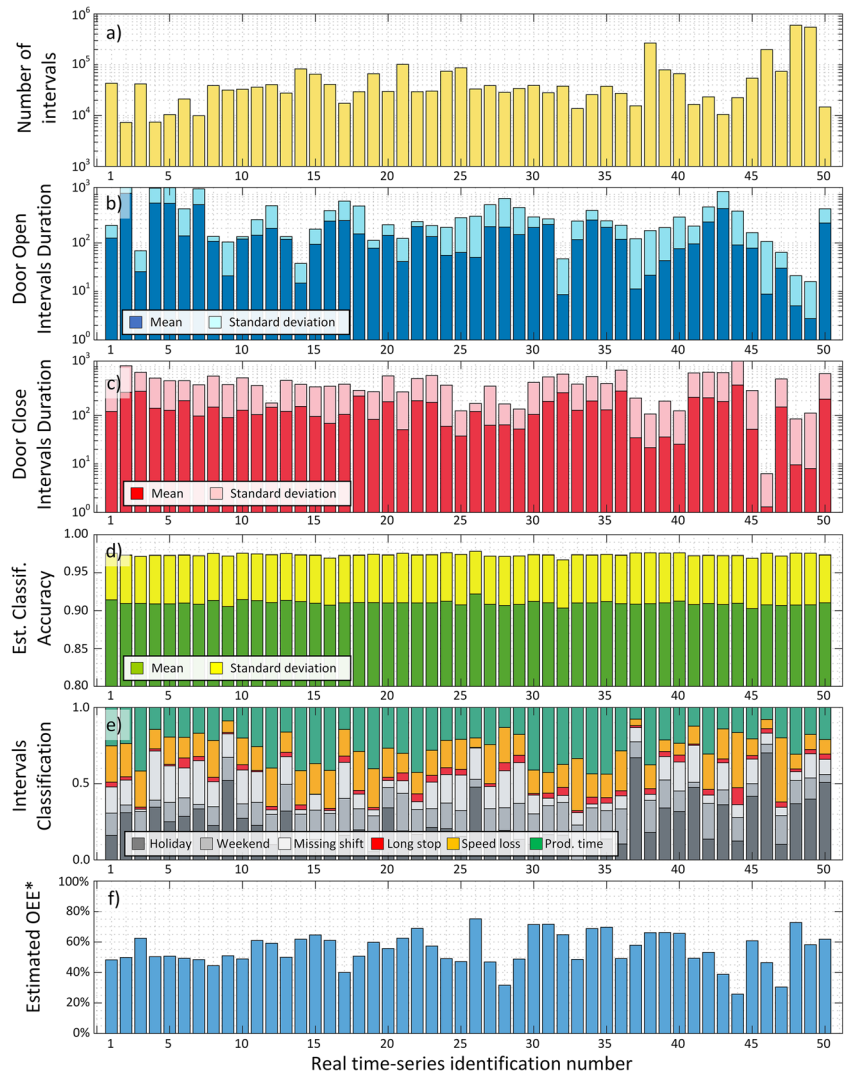
and

$$\sigma_{EBA}(i) = \sqrt{\frac{\sum_{j=1}^{18,695} \frac{BA^2(j) - \mu_{EBA}^2(i)}{d_{tss=j,tsr=i}}}{\sum_{j=1}^{18,695} \frac{1}{d_{tss=j,tsr=i}}}} \quad (17)$$

(16)

Since the balanced accuracy was high for all synthetic time series, the estimated accuracy was also rather high, with only

**Fig. 12** Final data description for the real time series: (a) total number of intervals composing the series; (b) mean and standard deviation of the open intervals; (c) mean and standard deviation of the close intervals; (d) mean and standard deviation of the estimated classification accuracy; (e) interval classification; (f) estimated  $OEE^*$ ; in charts (b), (c), and (d) standard deviations are stacked on top of the mean values



minimal differences. The classification of times into the six classes is given in Fig. 12(e). The distribution of times is very different from machine to machine. For instance, machines 37 and 46 were stopped for the largest amount of time in the sampling interval. Long stops and speed losses were observed across all machines, which was expected given that the investigated plant was not optimized.

Finally, a variant of the *OEE* was calculated for all machines as a performance indicator, and it is shown in Fig. 12(f). Usually, *OEE* is calculated by evaluating three factors [16]: availability, performance, and quality. Availability takes into account all major time losses, such as setup time and unexpected maintenance operations. Performance is related to speed losses and minor production interruptions. Quality considers the fraction of produced goods that are effectively compliant with specifications. To calculate the *OEE*, companies require an effective production schedule, accounting of setup and maintenance operations, and quality reports. It is usually a rather demanding task to extract all this information from different data sources (ERP, MES, Maintenance Management System) and compute the *OEE*.

In the described approach, some data are missing, thus preventing the precise calculation of the *OEE*:

- No data about the production schedule were provided by the company and the production schedule was different for different machines and varied over time;
- No quality data were available.

Accordingly, only a partial calculation of *OEE*, which did not include quality, was feasible. This variant, denoted as *OEE\**, was defined as the ratio of the sum of repetitive intervals to the sum of intervals excluding presumed holidays, as follows:

$$OEE^* = \frac{\sum_{j \in \text{Repetitive}} x(j)}{\sum_{j \in} x(j)} \quad (18)$$

By looking at the numbers, *OEE\** ranged from 0.26 to 0.75, thus clearly distinguishing likely non-critical machines from potential bottlenecks. As mentioned, an accurate quantification of the actual *OEE\** for a detailed comparison was unfortunately not available. However, the company confirmed that the computed *OEE\** values were in line with expectations and provided a rather accurate representation of the internal organization of the production plant. Plant managers were quite impressed by the possibility of obtaining these estimates almost automatically with such inexpensive and non-invasive sensors and without the need to combine sensor data with data extracted from other sources.

Clearly, these *OEE\** values cannot be considered a substitute for the conventional calculation of *OEE* due to the missing quality data and other classification approximations.

Additionally, as stated in the “Introduction,” this approach can be applied only in a semiautomatic and highly repetitive production environment.

On the other hand, the possibility of effortlessly obtaining an almost real-time overview of production status may be of great value in many production environments.

## 4 Conclusions

In this research, a prototype IIoT device was developed and applied to monitor the status of the main door of 50 machine tools in a factory producing batches of automotive parts. A preliminary analysis of the data collected from different machines showed a high degree of heterogeneity and patterns of open-close interval durations that clearly characterized production activities.

An innovative statistical algorithm for automatic identification and classification of repetitive patterns of data was developed. It was conceived to be very efficient from a computational point of view and relying only on basic statistical principles. To validate the algorithm, synthetic data provided by a realistic simulator developed during this research were used. All the details concerning the classification algorithm and the simulator are reported in this manuscript.

The validation results were good, with a balanced accuracy greater than 90% and very low variance. In addition, a preliminary comparison revealed that the proposed methodology exhibits advantageous classification performance compared to conventional machine learning classification methodologies.

The classification algorithm was applied to real data to evaluate speed losses and efficacy, and a simplified version of the *OEE* indicator was computed. The classification results were realistic and consistent with factory management observations obtained using traditional methods, thus further proving the validity and usefulness of the novel approach.

These results proved the possibility of monitoring the efficiency of production plants using very simple and economic sensors that can be easily installed on machine tools without complex integration procedures that would require mechanical or electrical modifications.

Future research will focus on comparing the algorithm performance against advanced machine learning approaches to investigate differences in classifying complex patterns. At the same time, the algorithm will be further developed to enhance analytical effectiveness and provide a valuable tool for actively optimizing manufacturing cycles and processes. To this aim, two primary improvement directions are recommended. First, enhance the algorithm adaptability to effectively handle not only on-off signals but also statuses defined by multiple non-binary sensors, thereby making the algorithm completely independent of the specific

characteristics of the manufacturing plant. Second, upgrade data analysis techniques by implementing intelligent strategies to enhance accuracy in pattern classification and extract even more meaningful insights from the collected data. In this regard, refining the algorithm based on real data from different manufacturing plants is crucial for improving its predictive power, while integrating anomaly detection logic could help identify unusual patterns or events that may be handled separately. The so-obtained information can be used alone or together with other data from the plant monitoring system to actively optimize manufacturing scheduling and plant management.

**Acknowledgements** The Laboratory for Advanced Mechatronics (LAMA FVG) of the University of Udine, Italy, is gratefully acknowledged for its technical support. LAMA FVG is an international research center for product and process innovation where the three universities of Friuli Venezia Giulia (Italy) cooperate to promote research and development activities at the academic and industrial levels. Emanuele Vaglio is grateful for funding under the REACT EU Italian PON 2014–2020 Program - Action IV.4 - Innovation (DM 1062, 10/08/2021).

**Author Contributions** Marco Sortino: conceptualization, methodology, formal analysis, investigation, resources, validation, writing—original draft. Emanuele Vaglio: formal analysis, investigation, validation, writing—review and editing. All authors read and approved the final manuscript.

**Funding** Open access funding provided by Università degli Studi di Udine within the CRUI-CARE Agreement. The authors declare that no funds, grants or other support were received during the preparation of this manuscript.

## Declarations

**Competing interests** The authors declare no competing interests.

**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

- Tran MQ, Doan HP, Vu VQ, Vu LT (2023) Machine learning and iot-based approach for tool condition monitoring: a review and future prospects. *Measurement* 207:112351. <https://doi.org/10.1016/j.measurement.2022.112351>
- Er PV, Teo CS, Tan KK (2016) Approach towards sensor placement, selection and fusion for realtime condition monitoring of precision machines. *Mech Syst Signal Process* 68–69:105–124. <https://doi.org/10.1016/j.ymssp.2015.07.008>
- Abele E, Panten N, Menz B (2015) Data collection for energy monitoring purposes and energy control of production machines. *Procedia CIRP* 29:299–304. <https://doi.org/10.1016/j.procir.2015.01.035>
- Sihag N, Sangwan KS (2020) A systematic literature review on machine tool energy consumption. *J Clean Prod* 275:123125. <https://doi.org/10.1016/j.jclepro.2020.123125>
- Teiwes H, Blume S, Herrmann C, Rössinger M, Thiede S. (2018) Energy load profile analysis on machine level. *Procedia CIRP* 69:271–276. <https://doi.org/10.1016/j.procir.2017.11.073>
- O'Driscoll E, Kelly K, O'Donnell GE (2015) Intelligent energy based status identification as a platform for improvement of machine tool efficiency and effectiveness. *J Clean Prod* 105:184–195. <https://doi.org/10.1016/j.jclepro.2015.01.058>
- Sihag N, Sangwan KS, Pundir S (2018) Development of a structured algorithm to identify the status of a machine tool to improve energy and time efficiencies. *Procedia CIRP* 69:294–299. <https://doi.org/10.1016/j.procir.2017.11.081>
- SeEVERS JP, Johst J, Weiß T, Meschede H, Hesselbach J (2019) Automatic time series segmentation as the basis for unsupervised, non-intrusive load monitoring of machine tools. *Procedia CIRP* 81:695–700. <https://doi.org/10.1016/j.procir.2019.03.178>
- Hacksteiner M, Duer F, Ayatollahi I, Bleicher F (2017) Automatic assessment of machine tool energy efficiency and productivity. *Procedia CIRP* 62:317–322. <https://doi.org/10.1016/j.procir.2016.06.034>
- Petruschke L, Walther J, Burkhardt M, Luther M, Weigold M (2021) Machine learning based identification of energy states of metal cutting machine tools using load profiles. *Procedia CIRP* 104:357–362. <https://doi.org/10.1016/j.procir.2021.11.060>
- Ahmad I, Hee LM, Abdelrhman AM, Imam SA, Leong MS (2021) Scopes, challenges and approaches of energy harvesting for wireless sensor nodes in machine condition monitoring systems: a review. *Measurement* 183:109856. <https://doi.org/10.1016/j.measurement.2021.109856>
- Dobra P, JÓsvai J (2021) Enhance of oee by hybrid analysis at the automotive semi-automatic assembly lines. *Procedia Manuf* 54:184–190. <https://doi.org/10.1016/j.promfg.2021.07.028>
- Montgomery D, Runger G (2013) *Applied Statistics and Probability for Engineers*, 6th Ed., John Wiley & Sons
- Wong T-T, Luk W-S, Heng P-A (1997) Sampling with hammersley and halton points. *J Graphics Tools* 2(2):9–24. <https://doi.org/10.1080/10867651.1997.10487471>
- Torabi M (2021) Matlab hammersley sampling for design of experiments doe. [https://github.com/Mansourt/MATLAB\\_Hammersley\\_Sampling\\_for\\_Design\\_of\\_Experiments\\_DOE/releases/tag/v1.0](https://github.com/Mansourt/MATLAB_Hammersley_Sampling_for_Design_of_Experiments_DOE/releases/tag/v1.0). Accessed 14 Oct 2023
- Bhade S, Hegde S (2020) Improvement of overall equipment efficiency of machine by smed. *Mater Today Proc* 24:463–472. <https://doi.org/10.1016/j.matpr.2020.04.298>

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