International Journal of Information Management 46 (2019) 252–262

Contents lists available at ScienceDirect

International Journal of Information Management

journal homepage: www.elsevier.com/locate/ijinfomgt

Data analysis and feature selection for predictive maintenance: A case-study in the metallurgic industry

Marta Fernandes[a,*], Alda Canito[a], Verónica Bolón-Canedo[b], Luís Conceição[a], Isabel Praça[a], Goreti Marreiros[a]

[a] GECAD - Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development, Polytechnic of Porto, Porto, Portugal
[b] Laboratory for Research and Development in Artificial Intelligence (LIDIA), Computer Science Dept., University of A Coruña, A Coruña, Spain

ARTICLE INFO

Keywords:
Predictive maintenance
Data analysis
Feature selection
Rule-based model

ABSTRACT

Proactive Maintenance practices are becoming more standard in industrial environments, with a direct and profound impact on the competitivity within the sector. These practices demand the continuous monitorization of industrial equipment, which generates extensive amounts of data. This information can be processed into useful knowledge with the use of machine learning algorithms. However, before the algorithms can effectively be applied, the data must go through an exploratory phase: assessing the meaning of the features and to which degree they are redundant. In this paper, we present the findings of the analysis conducted on a real-world dataset from a metallurgic company. A number of data analysis and feature selection methods are employed, uncovering several relationships, which are systematized in a rule-based model, and reducing the feature space from an initial 47-feature dataset to a 32-feature dataset.

1. Introduction

The competitiveness of a company is more crucial than ever in the current economic panorama, being immensely influential to the company's ability to provide quality products at low prices. Machine maintenance, with its direct impact in machine downtime and production costs, is directly related to a manufacturing companies' ability to be competitive in terms of cost, quality and performance (Aboelmaged, 2014; Holmberg et al., 2010). Modern maintenance approaches intend to lower failure rates and improve production times but aren't widely applied yet (Holmberg et al., 2010), with smaller companies demonstrating a lower level of e-maintenance readiness (Aboelmaged, 2014). These modern techniques reflect a transition from corrective maintenance practices to more proactive ones: proactive maintenance has the advantage of fixing problems before they come into place, replacing parts after a certain level of deterioration has been identified, as opposed to fixing the fault after the fact (Muller, Marquez, & Iung, 2008). Proactive maintenance includes preventive maintenance and predictive maintenance.

Preventive maintenance consists in performing periodic inspections and other operations according to a predetermined schedule, usually based on time in service. However, this type of maintenance is imperfect, unreliable and costly (Hashemian & Bean, 2011; Selcuk, 2017). To achieve a fully proactive approach, preventive maintenance must be complemented with predictive maintenance. Moreover, companies would benefit from using predictive maintenance throughout the equipment's life cycle to detect the onset of degradation and equipment failure (Hashemian & Bean, 2011; Selcuk, 2017). Predictive maintenance indicates the correct time to perform maintenance; as a result, machines spend less time offline and components are changed only and when needed. Predictive maintenance performs both prediction and diagnosis of an equipment's condition, providing information about the nature of the problem, where it is occurring and why, and when an equipment failure is likely to happen (Selcuk, 2017).

Predictive maintenance techniques can be implemented through the monitorization of equipment combined with intelligent decision methods. Machine Learning and Data Mining techniques can be used to draw insights from the data and accurately predict outcomes to support decision-making and help organizations improve their operations and competitiveness (O'Donovan, Leahy, Bruton, & O'Sullivan, 2015; Selcuk, 2017; Wang, 2013). Machine Learning approaches commonly used for fault detection and diagnosis include Artificial Neural Networks (Tian, 2012; Zhang, Wang, & Wang, 2013), Support Vector Machines (Li et al., 2014; Susto, Schirru, Pampuri, McLoone, & Beghi, 2015) and Decision Trees (He, He, & Wang, 2013), among others. However, these approaches tend to focus on vibration and sound

* Corresponding author at: GECAD – Instituto Superior de Engenharia do Porto, Rua Dr. António Bernardino de Almeida, no. 431, P-4249-015, Porto, Portugal.
E-mail address: mmdaf@isep.ipp.pt (M. Fernandes).

https://doi.org/10.1016/j.ijinfomgt.2018.10.006
Received 8 May 2018; Received in revised form 10 October 2018; Accepted 11 October 2018
Available online 30 October 2018

Nomenclature

cl_{m_i}	A machine's theoretical maximum coolant level
cl_{value}	A machine's current coolant level
id_{m_i}	A machine's unique identifier
id_{wp_i}	A workpiece's unique identifier
id_{φ_i}	A message's unique identifier
$lcpt$	Last Complete Part Timer
M	A set of machines
m_i	An individual machine
max_{diff}	Maximum difference between the theoretical maximum coolant level and the real maximum coolant level
max_{margin}	Upper threshold of the coolant's low level definition
min_{margin}	Lower threshold of the coolant's low level definition
mg_{φ_i}	The importance of a message (notification, alert or alarm)
$M30_t$	Current M30 Parts Counter
$M30_{t-1}$	Previous M30 Parts Counter
pt_i	Shortest recorded production time for a given workpiece type
ppt_t	Current present part timer
P_{wp_i}	Set of machines involved in the production of a workpiece and respective expected production time
R	A set of rules
$Red.$	Estimated redundancy on a pair of variables
ref_{value}	Real maximum coolant level value
Rcp_{φ_i}	A message's list of recipients
$safety_{margin}$	Safety margin of the spindle's rate
$sample\_max$	Maximum coolant level registered
$spindle_{load}$	A machine's current spindle load
$spindle_{rpm}$	A machine's current spindle rotating rate
$spindle_{timer}$	Number of seconds the spindle rate has been above its maximum rating
sr_{m_i}	A machine's maximum spindle rotating rate
ss_{m_i}	A machine's maximum spindle speed
Tx_{φ_i}	A message's text
V_{φ_i}	A list of variables associated with the context of the message
Wp	A set of workpieces
wp_i	A workpiece
φ_i	A message
ϕ	A set of messages

analysis (Banerjee & Das, 2012; Zhang et al., 2013), but the collection and analysis of other parameters, such as event data, should also be considered (Lee, Lapira, Bagheri, & Kao, 2013).

Due to the current trend of automation and data exchange in industrial environments, assisted by the rise of the Internet of Things (IoT), a large quantity of operational data is now either available, or can be acquired with relative ease. This can be done by interfacing with legacy systems and sensor networks and applying principles of IoT and Cyber Physical Systems (Al-Fuqaha, Guizani, Mohammadi, Aledhari, & Ayyash, 2015; Lee, Jin, & Bagheri, 2017). One of the big challenges of Predictive Maintenance is the need to deal with the large quantities of heterogeneous data now available, the so-called Big Data.

The use of Big Data technologies in the manufacturing industry is a relatively new, but fast-growing research area, encompassing disciplines such as automation, information technology and data analytics (O'Donovan et al., 2015; Santos et al., 2017). Big Data technologies can help improve employee productivity, reduce operating costs, refine companies' internal processes and improve data management, among other benefits (Raguseo, 2018). Deriving knowledge from large volumes of data is unfeasible with traditional data analysis techniques, therefore, the use of new technologies and processes to gain insights from datasets that are diverse, complex, and of a massive scale is essential to go beyond the state-of-the-art in predictive maintenance (Hashem et al., 2015; Lee, Jin, & Liu, 2017). Moreover, the analysis of Big Data in industrial contexts requires specific knowledge of the domain (Lee, Jin, Liu, 2017).

In this paper, we present our findings from the preliminary analysis of a data sample obtained from an ongoing Predictive Maintenance case-study in a metallurgical company. We describe the dataset and the insights gained from exploring the data. The analysis resulted in a

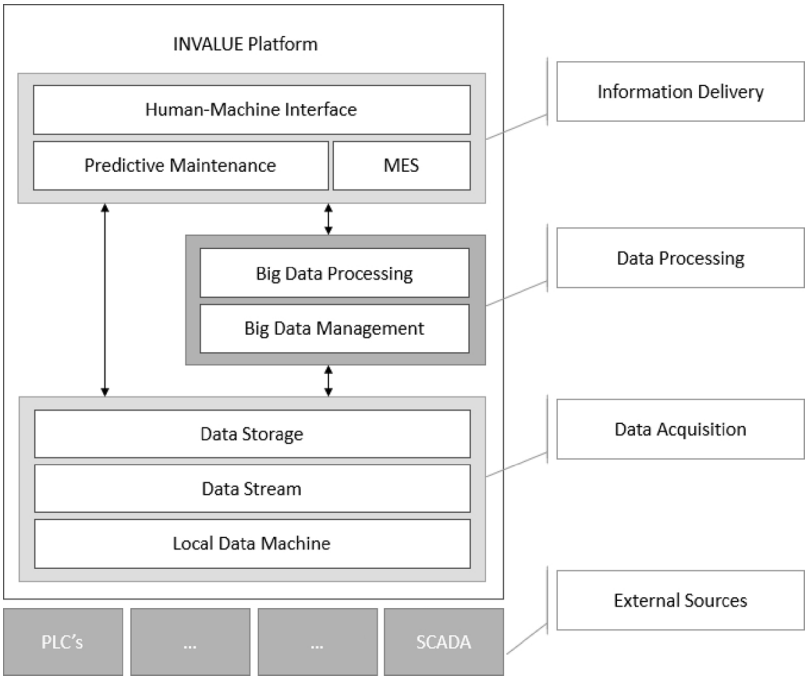

Fig. 1. InValue system's architecture.

reduction of the feature space and the conception of a rule-based model.

This document is organized as follows: (1) Introduction, wherein the theme and motivations were presented; (2) Background, where the architecture and analysed features are briefly explained; in Section (3), Data Analysis, the captured dataset is explored through several methods, followed by the application of (4) Feature Selection processes; these are followed by the specification of a set of rules which are presented in (5) Rule-based Model, leading to the (6) Discussion and, finally, (7) Conclusions.

2. Background

The Industrial Enterprise Asset Value Enablers project (InValuePT, 2017) aims to provide a platform that facilitates the shift from traditional maintenance approaches to more proactive ones. All stages the data must go through are addressed: from the acquisition processes, to its processing and subsequent delivery to the end users. This paper describes the preliminary analysis and features' selection that were performed on a dataset containing machining information from a metallurgical company. This data was gathered by the InValue system's acquisition module and concerns the operations performed by one of the company's lathe machines. These processes aim to uncover important relationships in the existing data, thus extracting relevant knowledge for the implementation of predictive maintenance approaches.

The InValue system is set up in a metallurgical company that is specialized in precision parts production and uses raw materials, such as aluminium, steel, bronze and technical plastics, to produce custom parts for industry clients.

The system's architecture comprises three main layers: (1) Data acquisition, (2) Data Processing and (3) Information delivery (Fig. 1). A more in-depth description of the architecture can be found in (Canito et al., 2017).

Performing Predictive Maintenance requires the system to monitor the manufacturing machines and obtain vast amounts of operating data. This task is performed by the data acquisition layer, which collects data from the machines and from the production management software. The machines facilitate information through a bus with a described protocol, which is consumed through a gateway. In case such a protocol is not defined, or terminals to existing sensors cannot be developed, new, external sensors, will first be installed.

All the data captured by the data acquisition layer is stored in a repository and made available to the other modules through a data stream. The repository can be queried directly by both the information delivery tier, mainly for visualization purposes, and by the Big Data processing module for data analysis and creation of predictive models. However, while creating the predictive models requires historical data to predict faults in the machines before they occur, such models must be used with real-time data. As such, the Big Data processing module also consumes the stream made available by the data acquisition module. The Big Data management module is responsible both for pre-processing the data and for employing Machine Learning and Data Mining techniques with the purpose of identifying components that might be approaching failure, diagnosing failures, and proposing possible corrective measures. The data is also analysed with the aim of suggesting actions that will lead to a decrease in waste production and a reduction in energy consumption.

For predictive maintenance to be carried out, it is imperative that the knowledge acquired by analysing the data reaches the right people at the right time. The company's collaborators will be able to visualize information that is pertinent to their specific functions and responsibilities, such as short-term alarms and notifications for machine operators and key-performance indicators for upper management employees. It will also be possible to view comparative analysis of similar equipment and conduct analytical monitoring of the processed data from different temporal perspectives. Furthermore, the proposed system will be integrated with the company's production and management software to aid the manufacturer improve their processes and reduce costs and maintenance times.

A prototype of the data acquisition module has been installed on one of the company's lathes, specifically a Haas ST-30 lathe (Haas Automation Inc, 2018). In a later stage of development, the system will monitor and collect data from four machines: two lathes and two vertical machining centres.

Table 1
Variables monitored by the data acquisition prototype.

Element	Description	Origin
Serial Number	Machine Serial Number	Machine Protocol
Control Software Version	Version of the machine's software	Machine Protocol
Machine Model Number	Machine's model number	Machine Protocol
Tool Changes (total)	Number of times a tool was changed since the machine was first powered on	Machine Protocol
Tool Number in Use	Turret station number currently in use	Machine Protocol
Dry Run	Indicates if the machine is running a program without producing a part	Machine Protocol
Power-On Time (total)	Time since the machine was powered on	Machine Protocol
Motion Time (total)	Time the machine is in motion	Machine Protocol
Last Cycle Time	Last production cycle time	Machine Protocol
Previous Cycle Time	Previous production cycle time	Machine Protocol
M30 Parts Counter #1	Counts the number of times a program completes.	Machine Protocol
M30 Parts Counter #2	Counts the number of times a program completes.	Machine Protocol
Maximum axis loads for X, Y, Z, A, B, C, U, V, W, T	Maximum load an axis has achieved since the machine was powered on	Machine Protocol
Coolant Level	Cutting emulsion level	Machine Protocol
Spindle load with Haas vector drive	Spindle load	Machine Protocol
Present part timer	Effective production time for the part currently in production	Machine Protocol
Last complete part timer	Effective production time for the part previously completed	Machine Protocol
Tool in spindle	Turret station number currently in use	Machine Protocol
Spindle RPM	Spindle rotation speed	Machine Protocol
Present machine coordinate position X, Y, Z, A, B	Current machine position for axes X, Y, Z, A, B	Machine Protocol
Present work coordinate position X, Y, Z, A, B	Position of the part at the start of production in axes X, Y, Z, A, B	Machine Protocol
Present Tool offset X, Y, Z, A, B	Distance of the tool relative to the origin in axes X, Y, Z, A, B	Machine Protocol
Machine Vibration X, Y, Z	Vibration during the cutting process on axes X, Y, Z	Sensor
Noise	Noise inside the machine	Sensor

The prototype collects machine and sensor data concerning forty-seven features, forty-three of which are obtained directly from the machine and four that are collected from external sensors. The acquisition of a total of seventy-eight features is planned for later stages of development, including critical data related to the machines' electrical components. The data would, ideally, include information about problems that occurred in the machine. However, due to circumstances beyond our control, the machines of the metallurgical company involved in the project fail very rarely. Consequently, that information is currently unavailable, which constrains the type of data analysis that can be performed.

A possible approach to this problem is the use of one-class classification methods, such as one-class SVM and autoencoder, to perform anomaly detection. Table 1 presents a small description of each of the variables monitored by the prototype.

3. Data analysis and correlation

The analysis presented here was performed on a sample of the collected data. Since the data was obtained from several sources, it was necessary to integrate and consolidate it prior to performing the analysis. The dataset consists of 23,134 rows and 48 columns, representing about 6 days of data acquisition. Each column represents the forty-seven features mentioned in Section 2, plus a column with the time/date at which the data was measured.

After analysing the sample for missing data, it became apparent that several features could be removed from the dataset:

- maximum axis load for axes Y, B, C, U, V, W, T;
- present machine coordinate position for axis Z;
- present work coordinate position for axis Z;
- present tool offset for axes Z, A, B.

The software installed in the lathe is the same across the entire product line. It is prepared to collect information about axes the ST-30 lathe doesn't have, but that might be present in other Haas machines. There are other features, such as the 'Tool Number in Use' and 'Tool in Spindle' that aren't very informative. This occurs because the value registered by the machine doesn't actually identify the tool in use, but the position in the turret where the tool has been placed. It is, therefore, impossible to identify the operating tool using only the information obtained from the machine.

A feature of interest is the 'Present Part Timer', which records the time it takes to effectively produce a given part. This means that time is only recorded when the part is actually being cut by the tool. Whenever the tool isn't actually operating on the part, the timer is paused. Because the acquisition of data happens continuously, this feature can be used to discern if a part is being produced or not. A relationship can be observed, in Fig. 2, between this feature and the feature 'Spindle Load with Haas Vector Drive'. When the line representing the 'Present Part Timer' is flat (which means the timer is paused), the spindle load is zero. When the tool is operating on a part it places a load on the spindle, so it's normal for the load to be zero when the tool isn't being used.

However, a part might not be completed successfully, a fact that cannot be perceived by analysing the "Present Part Timer" feature alone. More information can be obtained by looking at the "Last Complete Part Timer" and "M30 Parts Counter" variables, which can be used to complement the information provided by the "Present Part Timer". Most CNC programs end with code M30. This code signals the successful completion of a cycle and causes the program to reset, saving time during the mass production of parts. The variable "M30 Parts Counter" counts how many times the M30 code was executed, which is equivalent to how many parts were successfully produced. The "Last Complete Part Timer" records the time it took to produce a part whose program execution reached the M30 code. The first plot in Fig. 3 shows the "Last Complete Part Timer" data overlaid on the "Present Part Timer" and the second plot presents the "M30 Parts Counter". The red points in the first plot represent the moments when the value of the "Last Complete Part Timer" changed. As can be observed, if two or more consecutive parts take the exact same time to be produced the value of the "Last Complete Part Timer" won't change.

This information can be used to optimize the production of parts, since it shows when and for how long the machine is stopped and reveals variable production times for the same type of part. The time it takes to successfully complete a specific part, as given by the "Last Complete Part Timer", can be used as a reference value for the production of similar parts and can be updated whenever a part takes less time to complete. Any time the production time deviates negatively from that reference value, a notification or alarm can be issued. As mentioned above, the "Last Complete Part Timer" can't be used to check if a part was successfully produced, since its value will only change if a part takes a different time to produce. The "M30 Parts Counter", however, changes every time the production program of a part reaches the M30 code and can be used reliably to confirm a part

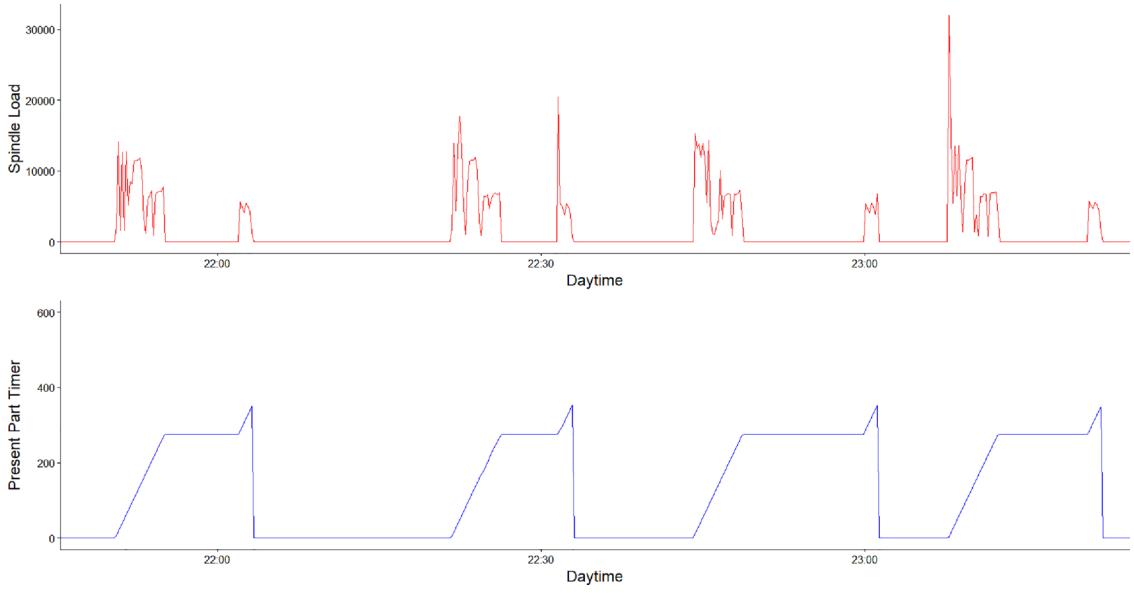

Fig. 2. *Part timer* and *spindle load* for the same time period.

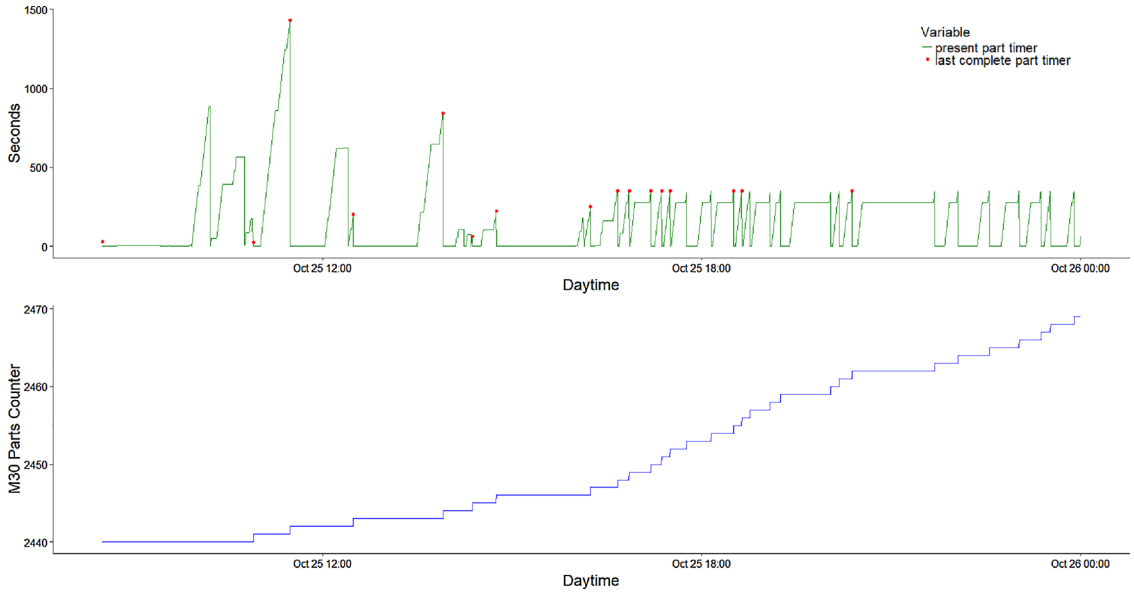

Fig. 3. Last Complete Part Timer overlaid on Present Part Timer and M30 Parts Counter. (For interpretation of the references to colour in the figure text, the reader is referred to the web version of this article.)

was successfully produced. This data can also be used to detect when problems occur during the production of a part. Fig. 4 shows that between 10:00 and 11:00 three parts were produced but not concluded, with a similar situation occurring after midday.

Information regarding the spindle, such as the spindle's load and speed, are very important since anomalies related with this component can be indicative of problems in the machine. The feature 'Spindle Load with Haas Vector Drive' records the spindle's load as the energy outputted by the vector drive to power the spindle's motion. The spindle has a maximum rating of 22.4 kW and a maximum speed of 3400 rpm. Theoretically, the machine is capable of working at this rate infinitely; however, it can sustain a load of 150% of the maximum capacity for 30 min, at most, and of 200% for 3 min, maximum. Fig. 5 shows changes in the spindle's load over the course of a day. It can be observed that the spindle worked below its maximum capacity for most of the day, but as both plots clearly show, it exceeded the 22.4 kW rate once. Further exploration revealed it happened for less than 10 s and didn't cause any problems. However, it is clear that the analysis of the spindle component provides valuable information for purposes of predictive maintenance. Fig. 6 shows the 'Spindle RPM' for the same period. The positive and negative values represent the clockwise and counter clockwise motion of the spindle, respectively.

The features representing the axes' coordinates can provide valuable information regarding the production of parts. The Haas ST-30 lathe provides information regarding the machine's coordinates, the workpiece's coordinates, and the tool offset coordinates. The machine's coordinates represent the machine's working plane in relation to the cutter's central point, while the workpiece's coordinates refer to the machine's position relative to the workpiece plane. Programming on a CNC machine uses the cutter's centre point as a reference, but different tools have different lengths and diameters, which, if not considered, will cause the tool to cut the wrong parts of the workpiece. Therefore, an offset value must be defined whenever a different tool is used. As such, the tool offset coordinates refer to the position of the tip of the

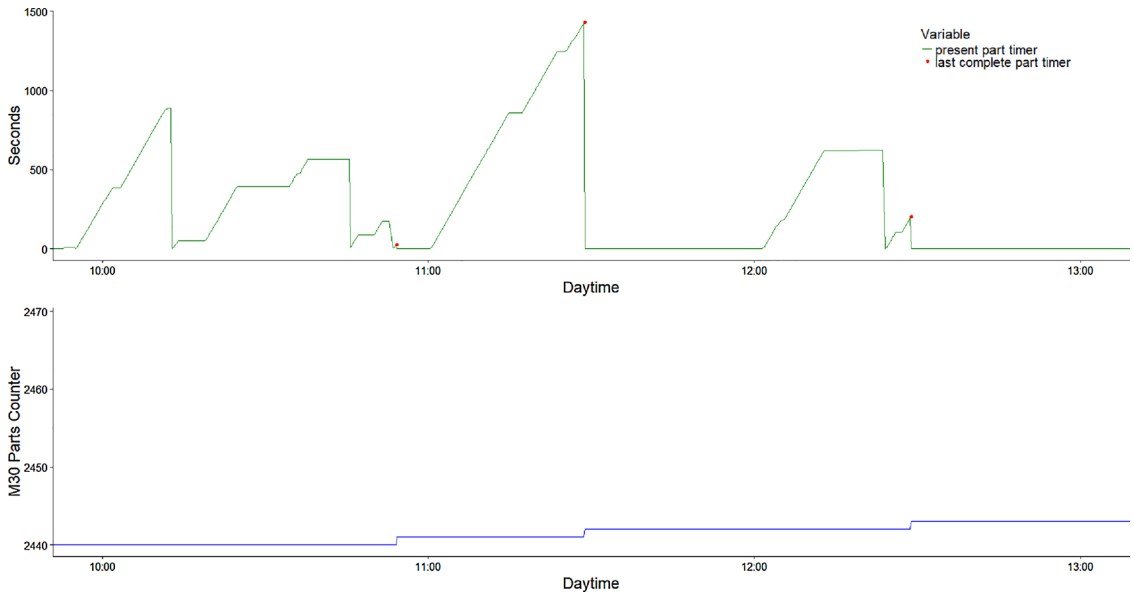

Fig. 4. Plot detail showing parts not successfully concluded.

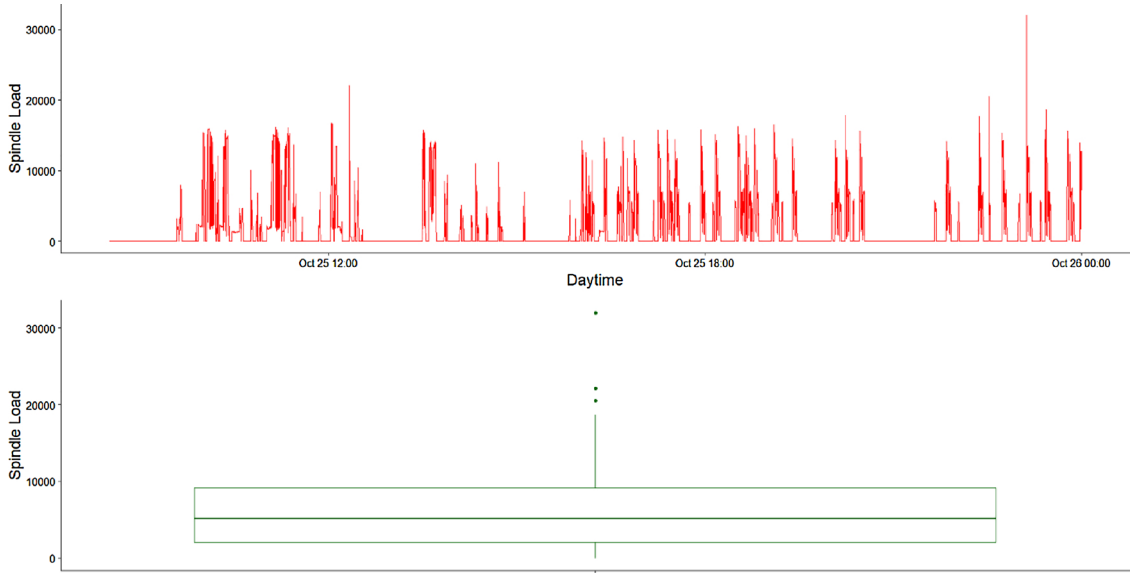

Fig. 5. Spindle load.

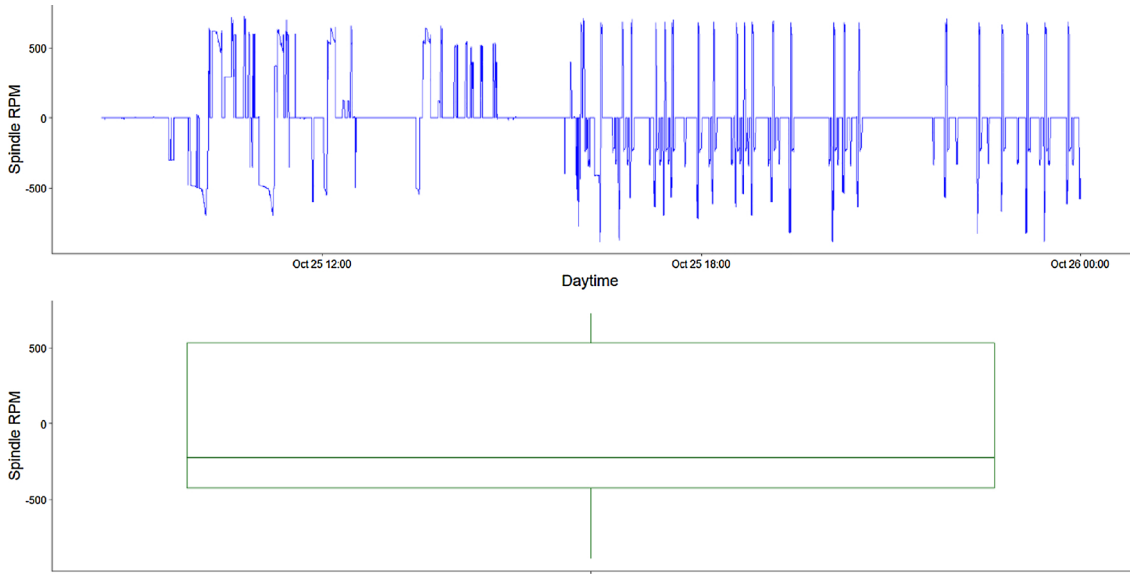

Fig. 6. Spindle RPM.

cutting tool relative to the cutter's centre point.

Every produced part has a specific design, meaning the production of different units of the same part should result in similar coordinate patterns. This pattern information can be used as input to machine learning algorithms to create models capable of distinguishing between normal and abnormal production of parts. Fig. 7 illustrates the production of several units of the same part. The continuous line demonstrates how the 'Machine Coordinate Position for Axis X' evolves over time, while the dashed line refers to the 'Present Part Timer' feature and shows when the production of each part began and ended. While similar, the patterns aren't exactly the same. Considering the different units in production have the exact same design, it would be expected for the coordinate patterns to also be the same, but they are not. This happens in part because the machine operator can intervene in the production of a part and perform manual operations, such as delaying or speeding up the production, which introduce variations to the expected pattern. Nevertheless, a coordinate pattern is still obvious and changes to that pattern represent anomalies in the production of a given part. This information can be related with other features, like the Spindle Load, to detect problems in the machine.

The coolant level is also a feature of interest, since this fluid plays an important role in the production of parts. Cutting a workpiece material generates a lot of heat, which can cause the tool to wear out rapidly, alter the metallurgical characteristics of the material and cause unwanted chemical reactions. As the name implies, the coolant, in this case a water-oil emulsion, serves the purpose of cooling the cutting tool and the workpiece. As such, information about the coolant level can be indicative of imminent problems in the machine (e.g. the coolant has run out) and can be related to other parameters to detect alterations in the machine's normal functioning. The data collected by monitoring this parameter can also be used to issue alarms and notifications regarding the current coolant level in order to prevent overheating problems.

The analysis of the data sample allowed us to understand which features are most important for detecting operational anomalies and how they can affect a machine's normal operation. It was possible to conclude that the features related to the spindle are crucial to detect problems in the machines, but information related to the coolant fluid

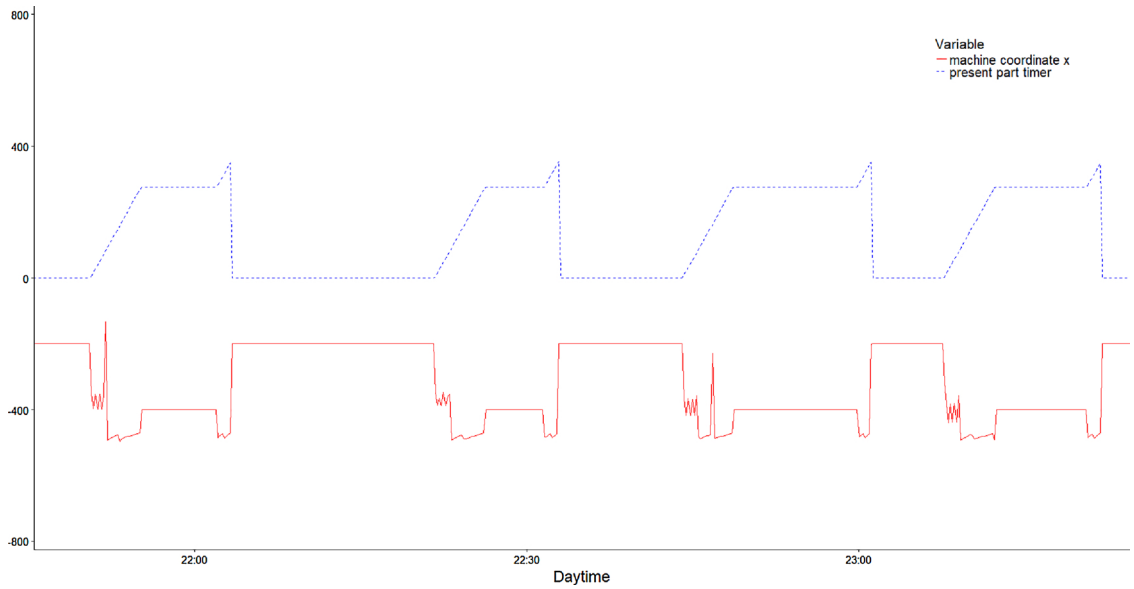

Fig. 7. Present Part Timer and Machine coordinate position for axis X during four production cycles.

and the machines' coordinates are also relevant. Additionally, the features concerning the production of parts provide valuable information to optimize their production. This information is relevant to the entire Haas product line, regardless of the context in which the machines are used and may even be applicable to similar machines built by other companies, provided that the necessary adaptations are considered.

4. Feature selection

With the advent of Big Data, datasets large in both number of instances and number of features are brought to the frontline of data analysis. The state space increases exponentially with the number of features, which then increases the computational requirements of most learning algorithms – for which the number of input features and sample size are important factors. Reducing the number of features of the problem is imperative to reduce both memory and computational requirements. The process of feature selection also improves the performance of many predictors by reducing overfitting and improving accuracy, among other benefits (Isabelle, 2006).

Feature selection (FS) methods typically belong to one of three classes, determined by the evaluation metric of choice: filter methods, wrapper methods and embedded methods (Chandrashekar & Sahin, 2014):

- *Filter methods* work as a pre-processing step and are independent of the learning algorithm, relying on the general characteristics of the training data. Their main advantages are the low computational cost and good generalization ability;
- *Wrapper methods* use the learning algorithm to measure the relative usefulness of subsets of variables. Because the number of subsets grows with the number of features and the algorithm has to be called for each one, these methods are prone to becoming computationally heavy;
- *Embedded methods* have the FS included in the learning process. The search for the optimal feature subset is built into the classifier and fulfils the role of searching the combined space of feature subset and hypothesis, capturing dependencies at a lower computational cost than wrappers.

In the context of this work, the method of feature selection that was applied is an adaptation of the Minimum Redundancy Maximum Relevance (mRMR) algorithm, which belongs to the category of filter methods. mRMR (Ding & Peng, 2005) ranks the importance of a set of features based on their relevance to the target, while penalizing features that are redundant among them. The goal of this algorithm is to find a subset of features which better describe the target class; in other terms, the subset of features the target class is most statistically dependent on. The complexity of this task increases with the dimensionality, particularly if the number of samples available is insufficient. As such, the maximum dependency calculation can be enhanced with two different criteria: *maximum relevance* and *minimum redundancy*.

Finding the maximum relevance means identifying the subsets of features that best describe the target class, that is, those with the greatest mutual dependence between the features and the target class. However, as described in (Ding & Peng, 2005), selecting the n most relevant features doesn't necessarily translate into the most descriptive subset: frequently, the n strongest features are also highly correlated, i.e., redundant. As such, the second criterion comes into play: that of *minimum redundancy,* which aims to find the n most distinct features in the feature-set.

As mentioned in Section 2, no data concerning actual failures is currently available (i.e., the target class is not known). Therefore, the mRMR algorithm had to be modified to run in an unsupervised fashion. Although it is not possible to compute the relevance between each feature and the class label, the criterion of *minimum redundancy* can be applied to select the subset of features with the lowest pairwise correlation. The criterion of minimum redundancy through mutual information was computed by normalizing the values in the dataset to the [0–1] range, further discretizing them in 5 intervals by employing the equal width binning strategy, which resulted in a 47×47-sized matrix of redundancies.

Of these, the 10% most redundant pairs were considered for further analysis, resulting in the 15 feature pairs shown in Table 2.

Several features are very closely related, and thus a higher redundancy was expected. That is the case of 'MP30PC1' and 'MP30PC2', which are both counters, counting how many times a given operation was executed; they happen to carry similar information on this particular dataset, but such is not a mandatory situation and they can be used to record different indicators. 'Tool in Use' and 'Tool in Spindle' have similar meanings and are updated under similar circumstances; 'Last Complete Part Timer' and 'Last Cycle Time' both refer to the time the machining process was on for a given part – albeit in different formats –, and thus a high redundancy is not surprising.

Considering the redundancy between 'Tool in Spindle' and 'Tool

Table 2
15 most redundant feature pairs.

Feature 1	Feature 2	Red.
MP30PC1	MP30PC2	2.2327
Tool in Spindle	Tool Number in Use	2.1321
Last Complete Part Timer	Last Cycle Time	1.9991
Present Machine coordinate Pos A	Present Work coordinate Pos A	1.9444
Tool in Spindle	Present Machine coordinate Pos A	1.8258
Tool in Spindle	Present Work coordinate Pos A	1.8247
Tool Number in Use	Present Machine coordinate Pos A	1.8226
Tool Number in Use	Present Work coordinate Pos A	1.8223
MP30PC1	Total Tool Changes	1.8056
MP30PC2	Total Tool Changes	1.8056
Present Work coordinate Pos X	Present Work coordinate Pos Y	1.1861
Present Machine coordinate Pos X	Present Work coordinate Pos X	1.0914
Last Complete Part Timer	Previous Cycle Time	1.0908
Last Cycle Time	Previous Cycle Time	1.0876
Coolant Level	Total Tool Changes	0.9294

Number in Use' described above, it follows that they both share close relationships with the same features. Interesting levels of redundancy among coordinates were discovered, both between coordinates and in relation to the 'Tool Number in Use'. Because the last is not a particularly informative feature, not much knowledge can be extracted from this relationship.

The counters 'MP30PC1' and 'MP30PC2' share some redundancy with the 'Total Tool Changes', considering that the tool can be changed after a work is completed, but this is not a requirement. The same rationale can be applied to the relationship between 'Coolant Level' and 'Total Tool Changes'.

The uncovered relationships allow us to reach some conclusions about which features could be excluded. 'Tool in Spindle' and 'Tool Number in Use' carry little and repeated information, and therefore only one of them needs to be kept; the same logic can be applied to the 'Last Complete Part Timer' and 'Last Cycle Time' pair. 'Last Cycle Time' and 'Previous Cycle Time' represent the same information, but in different moments in time. Since the acquisition process happens continuously, the data is bound to become duplicated and, consequentially, only one of them needs to be monitored. Work coordinates and Machine coordinates both refer to axial coordinates, but the axes may have different origin points. This, however, does not happen similarly for all coordinate pairs and therefore further exploration is required. As for different axes, they are independent features and thus excluding any of them is excluding valuable information.

The results of this algorithm show some relationships between features that were to be expected but reinforce the conclusions reached by the data analysis process, allowing for a reduction of the feature space from 47 to 32 features. These findings have also been validated by the machine's manufacturers.

5. Rule-based model

As described in Section 2, the InValue platform will employ Machine Learning and Data Mining techniques to uncover new knowledge about the operation and maintenance of equipment from the data collected by the Data Acquisition module. These Machine Learning models will be enhanced by a set of rules that provide information about the condition of specific operational parameters with the purpose of preventing deterioration of a machine's components and maximizing productivity.

These rules comprise the rule-based model presented in this section and were derived from the information and associations identified during the analysis described in Section 3.The model defines the actions the InValue system must take when certain conditions are met. It consists of a set of machines M and a set of workpieces Wp, such that a workpiece $wp \in Wp$ is associated with a set of machines and respective expected production times. The model also features a set of messages Φ and a set of rules R that will define the system's behaviour, as described below:

Definition 1. A rule-based model $(M,\ \Phi,\ R)$ consists of:

- a set of machines $M = \{m_1,\ m_2,\ ..., m_n\},\ n > 0$;
- a set of workpieces $Wp = \{wp_1,\ wp_1,\ ..., wp_1\},\ n > 0$;
- a set of messages $\phi = \{\varphi_1,\ \varphi_2,\ ..., \varphi_k\}$, k > 0;
- a set of rules $R = \{r_1,\ r_2,\ ..., r_j\},\ j > 0$.

Definition 2. A machine $m_i = \{id_{m_i},\ sr_{m_i},\ ss_{m_i},\ cl_{m_i}\}$ consists of:

- $\forall\ m_i \in M,\ i \in \{1,\ 2,\ ...,n\}$;
- id_{m_i} is the identifier of a given machine;
- sr_{m_i} is the maximum rating of the spindle of a given machine;
- ss_{m_i} is the maximum speed of the spindle of a given machine;
- cl_{m_i} is the maximum theoretical value of a given machine's coolant level.

Definition 3. A workpiece $wp_i = \{id_{wp_i},\ P_{wp_i}\}$ consists of:

- $\forall\ wp_i \in Wp,\ i \in \{1,\ 2,\ ...,n\}$;
- id_{wp_i} is the identifier of a given workpiece;
- P_{wp_i} is the set of machines the workpiece is produced in and respective expected production times, $P_{wp_i} = \{\{m_1,\ pt_1\},\ \{m_2,\ pt_2\},\ ..., \{m_n,\ pt_n\}\},\ n > 0$.

Definition 4. A message $\varphi_i = \{id_{\varphi_i},\ m_i, Rcp_{\varphi_i}, Tx_{\varphi_i}, V_{\varphi_i}, mg_{\varphi_i}\}$ consists of:

- $\forall\ \varphi_i \in \Phi, i \in \{1,\ 2,\ ...,n\}$;
- id_{φ_i} is the message's identifier;
- m_i is the machine associated with the message;
- Rcp_{φ_i} is a list of recipients, $\forall\ \varphi_i \in \phi,\ i \in \{1,\ 2,\ ...,n\},\ |Rcp_{\varphi_i}| > 0$;
- Tx_{φ_i} is the text associated with the message;
- $V_{\varphi_i}$is a list of variables associated with the context of the message;
- $mg_{\varphi_i}$is the importance of the message (Notification, Alert or Alarm).

Definition 5. The rules that make up the model define the relationships between variables uncovered in Section 3, and will describe the system's behaviour when certain conditions are met.

No domain information is available regarding the expected production time of a given part. As such, this value is initially determined by analysing the collected data, specifically the "Last Complete Part Timer" feature, and finding the shortest time it took to produce a part. The following rule updates that value whenever a part of the same type takes less time to produce:

Rule r1. $\forall\ P_{wp_i} \in Wp,\ i \in \{1,\ 2,\ ...,n\}\ ,\ whenever\ lcpt < pt_i\ then\ pt_i = lcpt$

Let us now define a rule that triggers an alert message anytime a part takes longer to produce than expected:

Rule r2. $\forall\ P_{wp_i} \in Wp,\ i \in \{1,\ 2,\ ...,n\}\ ,\ whenever\ lcpt > pt_i\ then\ send\ message\ \varphi_i$

$$= \begin{Bmatrix} id_{\varphi_j},\ m_i,\ [machine\ operator,\ production\ manager],\ "The\ current\ part\ took\ longer\ to \\ produce\ than\ expected.\ ",\ [last\ complete\ part\ timer,\ pt_i],\ Alert \end{Bmatrix}$$

The following rule causes the relevant people to be notified whenever the production of a part doesn't end successfully.

Rule r3. $\forall\ time\ instant\ t,\ whenever\ ppt_t = 0\ and\ M30_t = M30_{t-1}\ then\ send\ message\ \varphi_i$

$$= \begin{Bmatrix} id_{\varphi_j},\ m_i,\ [machine\ operator,\ production\ manager],\ "The\ part\ being\ produced\ wasn't \\ successfully\ concluded.\ ",\ [present\ part\ timer,\ M30\ parts\ counter\ ],\ Notification \end{Bmatrix}$$

The next two rules start and stop a timer every time the spindle load is greater or smaller than its maximum rating, respectively:

Rule r4. $\forall\ m_i \in M,\ i \in \{1,\ 2,\ ...,n\},\ whenever\ spindle_{load} > sr_{m_i}\ and\ spindle_{timer} = 0\ then\ start\ spindle_{timer}$

Rule r5. $\forall\ m_i \in M,\ i \in \{1,\ 2,\ ...,n\},\ whenever\ spindle_{load} \leq sr_{m_i}\ and\ spindle_{timer} > 0\ then\ stop\ spindle_{timer}$

Let us now define a rule that triggers an alert message when the spindle load exceeds the maximum rating:

Rule r6. $\forall\ m_i \in M,\ i \in \{1,\ 2,\ ...,n\},\ whenever\ spindle_{load} > sr_{m_i}\ then\ send\ message\ \varphi_i =$

$$\begin{Bmatrix} id_{\varphi_j},\ m_i,\ [machine\ operator],\ "The\ spindle\ is\ working\ at\ a\ higher\ rating\ than\ the\ maximum \\ recommended\ value.\ ",\ [spindle_{load},\ sr_{m_i}],\ Alert \end{Bmatrix}$$

The following rule triggers an alarm message if the spindle load exceeds the maximum rating up to 150% by more than 30 min:

Rule r7. $\forall\ m_i \in M,\ i \in \{1,\ 2,\ ...,n\},\ whenever\ spindle_{load} > sr_{m_i}\ and\ spindle_{load} \leq sr_{m_i} \times 1.5\ and\ spindle_{timer} \geq 1800\ then\ send\ message\ \varphi_i$

$$= \begin{Bmatrix} id_{\varphi_j},\ m_i,\ [machine\ operator],\ "The\ spindle\ has\ exceeded\ the\ maximum\ rating \\ and\ has\ been\ working\ at\ this\ rate\ for\ more\ than\ 30min.\ It\ must\ be\ brought\ down \\ to\ normal\ levels.\ ",\ [spindle_{load},\ sr_{m_i}],\ Alarm \end{Bmatrix}$$

The following rule triggers an alarm message if the spindle load exceeds the maximum rating between 150% and 200% by more than 3 min:

Rule r8. $\forall\ m_i \in M,\ i \in \{1,2,\ ...,\ n\},\ whenever\ spindle_{load} > sr_{m_i} \times 1.5\ and\ spindle_{load} \leq sr_{m_i} \times 2\ and\ spindle_{timer} \geq 180\ then\ send\ message\ \varphi_i$

$$= \begin{Bmatrix} id_{\varphi_j},\ m_i,\ [machine\ operator],\ "The\ spindle\ has\ greatly\ exceeded\ the\ maximum\ rating \\ and\ has\ been\ working\ at\ this\ rate\ for\ more\ than\ 3min.\ It\ must\ be\ brought\ down\ to \\ normal\ levels.\ ",\ [spindle_{load},\ sr_{m_i}],\ Alarm \end{Bmatrix}$$

The following rule triggers a message that will alert the machine operator any time the spindle exceeds 80% of the maximum speed:

Rule r9. $\forall\ m_i \in M,\ i \in \{1,\ 2,\ ...,n\},\ safety_{margin} = 0.8,\ whenever\ spindle_{rpm} > ss_{m_i} \times safety_{margin}\ then\ send\ message\ \varphi_i$

$$= \begin{Bmatrix} id_{\varphi_j},\ m_i,\ [machine\ operator],\ "The\ spindle\ speed\ is\ reaching\ a\ critical\ value.\ ", \\ [spindle_{rpm},\ ss_{m_i}],\ Alert \end{Bmatrix}$$

The following rule triggers an alarm message whenever the spindle exceeds the maximum speed:

Rule r10. $\forall\ m_i \in M,\ i \in \{1,\ 2,\ ...,n\},\ whenever\ spindle_{rpm} > ss_{m_i}\ then\ send\ message\ \varphi_i$

$$= \begin{Bmatrix} id_{\varphi_j},\ m_i,\ [machine\ operator],\ "The\ spindle\ has\ exceeded\ the\ maximum\ speed.\ ", \\ [spindle_{rpm},\ ss_{m_i}],\ Alarm \end{Bmatrix}$$

The coolant level has a maximum theoretical value, but in practice the maximum value varies, depending on the sensor installed in each machine. The only information available is that the real maximum value is close to the theoretical value and the coolant level is considered to be low once it's less than 50% of the maximum value. As such, it is necessary to empirically define a machine's maximum coolant level. This is done by finding the highest value of coolant level in the data sample and comparing it with the theoretical value.

Let us now specify a rule that defines the coolant level's maximum reference value in function of the theoretical value and the sample's maximum value:

Rule r11. $\forall\ m_i \in M,\ i \in \{1,\ 2,\ ...,n\},\ max_{diff} = 0.1,\ \exists\ ref_{value} = \begin{cases} cl_{m_i},\ sample\_max < cl_{m_i} \times max\_diff \\ sample\_max,\ sample\_max \geq cl_{m_i} \times max\_diff \end{cases}$

The following rule will trigger an alert message if the current coolant level is between 50% and 60% of the reference value:

Rule r12. $\forall\ m_i \in M,\ i \in \{1,\ 2,\ ...,n\},\ max_{margin} = 0.6,\ min_{margin} = 0.5,\ whenever\ cl_{value} > ref_{value} \times min_{margin}\ and\ cl_{value} < ref_{value} \times max_{margin}\ then\ send\ message\ \varphi_i$

$$= \{id_{\varphi_j}\ m_i,\ [machine\ operator],\ "The\ coolant\ level\ is\ getting\ low.\ ",\ [cl_{value},\ cl_{m_i}],\ "Alert"\}$$

The following rule will trigger an alarm message if the coolant level's current value is lower than 50% of the reference value.

Rule r13. $\forall\ m_i \in M,\ i \in \{1,\ 2,\ ...,n\},\ min_{value} = 0.5,\ whenever\ cl_{value} \leq ref_{value} \times min_{value}\ then\ send\ message\ \varphi_i$

$$= \begin{Bmatrix} id_{\varphi_j},\ m_i,\ [machine\ operator],\ "e;\ The\ coolant\ level\ is\ low.\ \ Replenish\ immediately!"e;, \\ [cl_{value},\ cl_{m_i}],\ Alarm \end{Bmatrix}$$

6. Discussion

While the InValue system encompasses several stages, from the acquisition of data to the delivery of information, the authors' main objective is analysing the data and building models capable of predicting faults in the machines. A considerable obstacle to that objective is the absence of faults, which constrains the type of data analysis that can be performed.

A sample of the collected data has been analysed with the aim of better understanding the data and identifying the most significant features and relationships. Whereas the analysis was performed on a relatively small sample of data, the insights drawn from the data are applicable to a much larger sample size and are supported by the domain knowledge provided by the machine's manufacturers. Data concerning the spindle was found to be of particular importance to detect problems in the machines, and it was also possible to uncover patterns of machine coordinates related to the machining of specific parts that can be used as input in the detection of anomalies. Moreover, insights were obtained that can contribute to optimize the production of parts and signal problems during the cutting process.

Additionally, a Feature Selection method was employed in order to assess redundancy between pairs of features. Although many of the discovered relationships were to be expected and support the conclusions reached by the exploratory analysis, some of them are interesting and surprising, requiring further study. Our analysis allowed us to start from a 47-feature dataset and scale it down to a 32-feature dataset, as is demonstrated in Table 3.

As shown, the reduction reflects the exclusion of features related to axes that are not used by the machine in the study, features that represent the same information but in different points in time and features concerning the same information but with different data types.

The knowledge acquired by analysing the data has also led to the definition of a rule-based model, which specifies the relations between the variables of interest, the machines and the actions taken by the InValue system.

7. Conclusions

This paper focused on the problem of carrying out predictive maintenance in a metallurgical company and presented the results of the preliminary data analysis and feature selection that were performed on a sample of the collected data. The insights derived from the data

Table 3
The 32 features considered more relevant according to the analysis.

Serial Number	Dry Run	Last complete part timer	Maximum axis loads for X, Z, A
Control Software Version	Power-On Time (total)	Spindle RPM	Present machine coordinate position X, Y, A, B
Machine Model Number	Motion Time (total)	Coolant Level	Present work coordinate position X, Y, A, B
Tool Changes (total)	M30 Parts Counter #1	Spindle load with Haas vector drive	Present Tool offset X, Y
Tool Number in Use	M30 Parts Counter #2	Present part timer	Machine Vibration X, Y, Z
Noise			

will assist in the development of adaptive learning models capable of handling complex information that can be applied to an entire product line of industrial equipment. Additionally, a number of rules were extracted from the relationships found during the data analysis process that were consolidated in a rule-based model. These rules will be the foundation of a rule-based system, which will be used to complement the predictive models that will be generated in the future.

The next stage of this work will also include collecting data from additional variables, such as the system temperature and different electrical components. This step is crucial, since more information about the machines' operating status can be obtained from them. Considering that all Haas machines employ the same firmware, it is possible that similar conclusions could be achieved both in terms of relevant features and their meaning. Future experiments will involve the monitorization of other machines to establish if these conclusions can be extended to different equipment.

Monitoring industrial machines in real-time results in large amounts of data that can't be analysed using traditional methods. As such, future work will also include the use of Big Data technologies.

Declaration of interests

None.

Funding

The present work has been developed under the EUREKA - ITEA2 Project INVALUE (ITEA-13015), INVALUE Project (ANI|P2020 17990), and has received funding from FEDER (ERDF) Funds through NORTE2020 program and from Portuguese National Funds through FCT (Portuguese Foundation for Science and Technology) under the project UID/EEA/00760/2013. These programs were not involved in the collection, analysis or interpretation of data nor had any role in the decision to submit the article for publication.

References

Aboelmaged, M. G. (2014). Predicting e-readiness at firm-level: An analysis of technological, organizational and environmental (TOE) effects on e-maintenance readiness in manufacturing firms. *International Journal of Information Management, 34*(5), 639–651.

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials, 17*(4), 2347–2376.

Canito, A., Fernandes, M., Conceição, L., Praça, I., Santos, M., Rato, R., Cardeal, G., Leiras, F., & Marreiros, G. (2017). *An Architecture for proactive maintenance in the machinery industry. International Symposium on Ambient Intelligence.* Cham: Springer254262.

Banerjee, T. P., & Das, S. (2012). Multi-sensor data fusion using support vector machine for motor fault detection. *Information Sciences, 217*, 96–107.

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering, 40*(1), 16–28.

Ding, C., & Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology, 3*(2), 185–205.

Haas Automation Inc (2018). *Haas ST-30 lathe product specifications.* Retrieved from: . . (Accessed 30th July 2018) https://int.haascnc.com/mt_spec1.asp?intLanguageCode=1033&id=ST-30&webID=2AXIS_STD_LATHE.

Hashem, I. A. T., Yaqoob, I., Anuar, N. B., Mokhtar, S., Gani, A., & Khan, S. U. (2015). The rise of "big data" on cloud computing: Review and open research issues. *Information Systems, 47*, 98–115.

Hashemian, H. M., & Bean, W. C. (2011). State-of-the-art predictive maintenance techniques. *IEEE Transactions on Instrumentation and measurement, 60*(10), 3480–3492.

He, S. G., He, Z., & Wang, G. A. (2013). Online monitoring and fault identification of mean shifts in bivariate processes using decision tree learning techniques. *Journal of Intelligent Manufacturing, 24*(1), 25–34.

Holmberg, K., Adgar, A., Arnaiz, A., Jantunen, E., Mascolo, J., & Mekid, S. (Eds.). (2010). *E-maintenance.* Springer Science & Business Media.

InValuePT (2017). *InValuePT.* Retrieved from: . . (Accessed 21st November 2017) http://www.invalue.com.pt/.

Isabelle, G. (2006). *Feature extraction foundations and applications.* Pattern Recognition.

Lee, J., Jin, C., & Liu, Z. (2017). *Predictive big data analytics and cyber physical systems for TES systems. Advances in through-life engineering services*. Cham: Springer pp. 97–112.

Lee, J., Jin, C., & Bagheri, B. (2017). Cyber physical systems for predictive production systems. *Production Engineering, 11*(2), 155–165.

Lee, J., Lapira, E., Bagheri, B., & Kao, H. A. (2013). Recent advances and trends in predictive manufacturing systems in big data environment. *Manufacturing Letters, 1*(1), 38–41.

Li, H., Parikh, D., He, Q., Qian, B., Li, Z., Fang, D., et al. (2014). Improving rail network velocity: A machine learning approach to predictive maintenance. *Transportation Research Part C: Emerging Technologies, 45*, 17–26.

Muller, A., Marquez, A. C., & Iung, B. (2008). On the concept of e-maintenance: Review and current research. *Reliability Engineering & System Safety, 93*(8), 1165–1187.

O'Donovan, P., Leahy, K., Bruton, K., & O'Sullivan, D. T. (2015). Big data in manufacturing: A systematic mapping study. *Journal of Big Data, 2*(1), 20.

Raguseo, E. (2018). Big data technologies: An empirical investigation on their adoption, benefits and risks for companies. *International Journal of Information Management, 38*(1), 187–195.

Santos, M. Y., e Sá, J. O., Andrade, C., Lima, F. V., Costa, E., Costa, C., et al. (2017). A Big data system supporting Bosch Braga industry 4.0 strategy. *International Journal of Information Management, 37*(6), 750–760.

Selcuk, S. (2017). Predictive maintenance, its implementation and latest trends. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 231*(9), 1670–1679.

Susto, G. A., Schirru, A., Pampuri, S., McLoone, S., & Beghi, A. (2015). Machine learning for predictive maintenance: A multiple classifier approach. *IEEE Transactions on Industrial Informatics, 11*(3), 812–820.

Tian, Z. (2012). An artificial neural network method for remaining useful life prediction of equipment subject to condition monitoring. *Journal of Intelligent Manufacturing, 23*(2), 227–237.

Wang, K. S. (2013). Towards zero-defect manufacturing (ZDM)—a data mining approach. *Advances in Manufacturing, 1*(1), 62–74.

Zhang, Z., Wang, Y., & Wang, K. (2013). Fault diagnosis and prognosis using wavelet packet decomposition, Fourier transform and artificial neural network. *Journal of Intelligent Manufacturing, 24*(6), 1213–1227.