Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 10 (2017) 997 – 1008

Procedia MANUFACTURING

45th SME North American Manufacturing Research Conference, NAMRC 45, LA, USA

Machine learning-based CPS for clustering high throughput machining cycle conditions

Javier Diaz-Rozo[a,b]*, Concha Bielza[b], Pedro Larrañaga[b]

[a]*Plethora IIoT, San Antolin 3, Elgoibar 20870, Spain*
[b]*Department of Artificial Intelligence, Technical University of Madrid, Madrid, Spain*

Abstract

Cyber-physical systems (CPS) have opened up a wide range of opportunities in terms of performance analysis that can be applied directly to the machine tool industry and are useful for maintenance systems and machine designers. High-speed communication capabilities enable the data to be gathered, pre-processed and processed for the purpose of machine diagnosis. This paper describes a complete real-world CPS implementation cycle, ranging from machine data acquisition to processing and interpretation. In fact, the aim of this paper is to propose a CPS for machine component knowledge discovery based on clustering algorithms using real data from a machining process. Therefore, it compares three clustering algorithms —k-means, hierarchical agglomerative and Gaussian mixture models— in terms of their contribution to spindle performance knowledge during high throughput machining operation.

Peer-review under responsibility of the organizing committee of the 45th SME North American Manufacturing Research Conference

Keywords: clustering; cyber-physical system; IIoT; behavior pattern; knowledge discovery

1. Introduction

Nowadays, performance models in the high-throughput machine tool sector are developed mainly theoretically or under controlled conditions because data are not available during real operation. Data are in short supply because there are no data gathering systems and/or on the grounds of client confidentiality. Consequently, machine tool

* Corresponding author. Tel.: +34-647-737-113
E-mail address: jdiaz@plethoraiiot.com

Peer-review under responsibility of the organizing committee of the 45th SME North American Manufacturing Research Conference
doi:10.1016/j.promfg.2017.07.091

manufacturers only have access to information about their machines when a fault has occurred. As a result, they cannot retrieve any previous information useful for pinpointing the causes of the failure.

The cyber-physical system (CPS) concept has been developed under the umbrella of the Industrial Internet of Things (IIoT) to integrate computational and physical capabilities and provide useful information for operators [1]. This integration relies on the fact that the data produced by a machine are useful for understanding and fine-tuning the theoretical models (e.g., finite element analysis). Additionally, it gives the designers or maintenance personnel information about failures. As a result, CPS are now integrated into machine tools as a performance monitoring tool.

At machine level, CPS enable the use of data coming from different components to gather useful knowledge. However, the main efforts within data driven systems (DDS) are based on supervised approaches used to estimate the remaining useful life (RUL) of a ball-bearing using band energy measurements [3] or on more advanced techniques like Kalman Filters [4], tool life calculation using dynamic Bayesian networks [5] and RUL estimation for peripheral components like elevator doors using logistic regression models [6]. Regarding unsupervised learning applications on real data, the main uses are in logistics, supply chain and production [7] for detecting poor yield factors based on self-organizing maps and the rule of induction [8], increasing yield in semiconductor manufacturing by k-means [9], improving the refinery catalytic process based on a k-means variation [10] and identifying welding flaws based on fuzzy clustering [11]. In terms of knowledge discovery, a framework for extracting useful knowledge from CAD event data has been developed applying a clustering algorithm that groups design task activities [12].

Therefore, this paper proposes the development of a component behavior multidimensional pattern using machine learning clustering techniques. Clustering algorithms aim to partition a dataset into clusters, interpreted as behavior patterns (operating modes or classes). Some properties of these clusters like the mean can give insight into their interpretation. To illustrate this approach, this paper focuses on the application of a CPS device to get process knowledge about the machine spindle during its operation. This process is called knowledge discovery [13], where clustered instances (an instance is characterized by variable values associated with the same timestamp) with common characteristics can provide useful information about spindle behavior during operation. This behavior could be detected or represented by the clusters in an unsupervised manner [14], where instances are partitioned based on metrics like Euclidean distances or probability distributions. Such metrics do not take into account the physical meaning of each variable leading to uncoupled knowledge discovery.

2. Methodology

One of the main objectives of this paper is to show how IIoT technologies have enabled the use of real data for diagnosis systems. Therefore, data for this experiment will be taken from a real machine tool, shown in Figure 1a. This machine produces one of the most important powertrain components inside car manufacturer production plants around the world. The throughput time is two components every 60 seconds with quality requirements of less than one rejected produced part per million. Specifically, this machine is integrated into a production line and is equipped with two spindles performing the same operation at the same time in order to process two components and accomplish the required production line cycle time.

Because of car manufacturer production requirements, machine utilization has to be high, normally over 95%. At this level, machine component degradation has to be closely controlled in order to minimize failures that may affect the overall line productivity. On this ground, specific maintenance procedures are required to care for critical parts like spindles. The objective of these procedures is to avoid unexpected failures during operation. This is known as preventive maintenance and is very expensive as components are replaced after a specified time in operation irrespective of their condition.

The spindle function, shown in Figure 1b, is to deliver the power needed by the tool to extract material from the workpiece, in this case, special powertrain steel. The power rating for the spindles equipped inside this machine is 10 kW, subjecting mechanical and electrical parts like bearings and windings to high stress levels. From our experience, spindle failure in this industrial sector may amount to USD 30,000 on a spindle spare, plus five maintenance work shifts to replace the spindle. As the production line has to be completely stopped, close to 200 direct operators are unable to work, increasing the economic impact on the company.

Fig. 1. (a) Machine tool used for the experiments and (b) spindle detail

Because of the aforementioned concerns, the development of a more precise diagnosis tool for machine spindle is a reasonable objective. Therefore, an embedded electronic-based CPS has been developed to gather data from both spindles during machining and then analyze the data using clustering algorithms to get relevant information.

2.1. Spindle operation data acquisition and pre-processing

The experimental setup shown in Figure 2 is proposed in order to implement the embedded electronic system into a machine-tool to gather the required data. In this case, data coming from the Computer Numeric Control/Programmable Logic Controller (CNC/PLC) unit via an Ethernet interface is gathered by the CPS. The CPS has bi-directional communication capabilities with the CNC and is able to request the machine state on which it needs to record data. Once data has been gathered and a timestamp has been assigned, the instance is sent to the local hard drive device (HDD) to be stored in comma separated value format (csv). Figure 2(a) shows the physical arrangement of the setup and Figure 2(b) shows its integration into the machine's electric cabinet.

The device shown in Figure 2(c), developed by Ikergune and SoC-e, is able to establish peer-to-peer communication with machine tool control system, i.e., programmable logic controller (PLC) and computer numeric control (CNC). This industrial grade CPS is a SoC-e CPPS-Gate40 system based on Zynq FPGA and equipped with different communication ports to get real-time data from the machine. The FPGA is able to provide some pre-processing facilities like filtering or changing the working domain (e.g., from time domain to frequency domain).

The machine tool is controlled by a SIEMENS Sinumerik 840D PLC/CNC which uses the Step7 protocol to send data through an Ethernet port to the CPS extracted directly from the internal CNC variables. The applicable procedure is to store the required variables in an internal database and send the package once all the variables are read. This technique guarantees extremely fast data communication without control system memory usage.

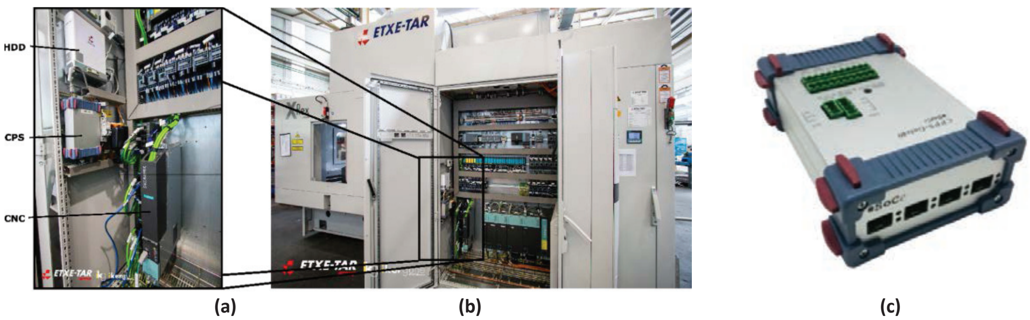

Fig. 2. (a) Detail of experimental setup, (b) machine tool integration, and (c) CPS hardware

Since the objective in this case is to analyze spindles during a machining process, the CPS needs to differentiate when the machine is cycling with a workpiece. In this manner, it is able to preprocess data and rule out unnecessary data that might embroil later analysis. Also, the CPS queries variables like actual time in order to synchronize the different systems involved in data acquisition. Once the acquisition system has been integrated, the CPS is programmed to gather the following variables from the machine's CNC: power consumption in kW, angular speed in RPM, torque in N·m and temperature in degrees Celsius. These process variables are important for describing and diagnosing electrical motors operation as described by [15].

Furthermore, other variables are taken from the PLC, as explained above: timestamp, machine and workpiece presence, signaling when the machining process is running and the cycle is not empty. These variables are used for synchronization and pre-processing purposes only.

In this case, the machine used in the analysis is able to process two workpieces at the same time in order to guarantee the required cycle time (around 60 seconds per part). To do this, it is equipped with two spindles that perform exactly the same machining operation on two workpieces, since it takes around 120 seconds to complete the machining process for both workpieces. The fact that both spindles perform exactly the same operation provides a perfect comparison framework. For this machine, variables are sampled at 10 Hz in order to get transitions during 10 machining cycles. Twenty workpieces are produced during the 10 machining cycles, 10 workpieces by each spindle.

2.2. Machine learning-based behavior pattern development

In order to find the spindle behavior patterns, we use unsupervised learning as there is no class to be predicted. Instead instances are partitioned into natural groups called clusters [16].

The algorithms are briefly explained below:

- **K-means clustering:** This algorithm divides a set of N d-dimensional instances into k clusters by minimizing the sum of the squared distances between the mean of the cluster (centroid) and the within-cluster points [17]. The algorithm starts with a selection of k initial centroids. Next, it assigns the closest instances to each cluster. The cluster centroids are then updated. The algorithm continues until there is no change (i.e., convergence) in the centroid values.
- **Agglomerative hierarchical clustering:** This algorithm uses a bottom-up or ascending approach based on the assumption that each instance is a cluster [18]. It then merges the pair of clusters with the smallest distance (or linkage) to create N-1 clusters. This process is iterated until a single cluster is formed containing all initial points.
- **Gaussian mixture model clustering:** This algorithm assumes that instance points have been generated by a mixture of Gaussian distributions with unknown parameters [19]. The unknown parameters are estimated by maximizing the log-likelihood, using the expectation-maximization algorithm. Therefore, an instance is assigned to cluster with some probability.

For comparison purposes, we used the Python-based implementation of these clustering algorithms in the Scikit-learn library [20]. Full implementation details are available at the Scikit-learn webpage[†].

Metrics, like the homogeneity score, completeness score, V measure, adjusted Rand index, adjusted mutual information, silhouette coefficient, are used to evaluate the performance of clustering algorithms. However, such metrics are not reliable enough [21]. Therefore, the results are evaluated based on the opinion of a machining expert in this paper.

Additionally, it is important to stress that results are sensitive to the clustering method used. Therefore, the expert evaluation is based on contribution to new knowledge of the spindle behavior.

3. Experimentation results

As described in Section 2, ten machining cycles were recorded using the CPS at 10 Hz, giving a total of 91,297 instances to be analyzed with clustering algorithms. Each instance contains power, torque and temperature data for each spindle. Therefore, data for each spindle is fed into the clustering algorithm and then instances are scattered in terms of their cluster label assignment. The results are reported below.

[†] http://scikit-learn.org/

3.1. K-means clustering

K-means has three main parameters: (a) number of clusters k, (b) initialization, and (c) distance. As regards the number of clusters k, the selected value after testing several values of k is based on spindle operating states: idle, acceleration/deceleration and full power, i.e., k = 3. With respect to initialization, k-means is sensitive to selection, yielding different results in each iteration, as it may converge to local optima. Therefore, there are different strategies based on the random choice of initial cluster centers or faster strategies that attach weights to initial instances depending on the squared distance to the centroid, called k-means++ [22]. K-means++ is the option selected in this paper. Finally, we use the Euclidean distance to find spherical or ball-shaped clusters in data [23] in order to get clusters that are as convex as possible and are easier for the expert to analyze. The other parameters used were left as Scikit-learn library defaults where they are available for consultation.

Figure 3 shows spindle torque versus power consumption. In this case, Cluster 0 is integrated by instances with negative torque from -33 to -8 N•m and the power range is from -2 kW to 10 kW. Cluster 1 is composed of instances that mainly have a torque between -8 and 8 N·m and a narrow power consumption range from -1 to 3 kW. Cluster 1 has instances with a positive torque range between 9 and 33 N·m with a power range from -10 kW to 2 kW. The shapes and the sizes for each cluster are similar for both spindles across these two dimensions with almost the same number of clustered instances.

The relation between temperature and power is shown in Figure 4. In this case, Cluster 0 and Cluster 2 have a temperature range from 35.5 to 36.5ºC. For Cluster 1, this temperature range is 33.8 to 36.5ºC. In these dimensions, cluster sizes and shapes are similar too. Table 1 summarizes centroid locations in each cluster dimension and the clustered instances per spindle.

3.2. Agglomerative hierarchical clustering

The parameters of this algorithm are: (a) linkage method, (b) number of clusters k, and (c) connectivity matrix. The linkage method is the distance between two clusters used by the algorithm for merging purposes. We use Ward's method, which minimizes the variance within clusters. The number of clusters k is chosen as in k-means clustering, with k = 3. Finally, with spatial data, the connectivity matrix is useful for telling the algorithm the neighboring positions of each instance. In this case, the matrix will be used exclusively to speed up the algorithm and avoid searching for neighborhoods. It is calculated previously by finding each instance's neighbors. Other parameters used were left as Scikit-learn library defaults.

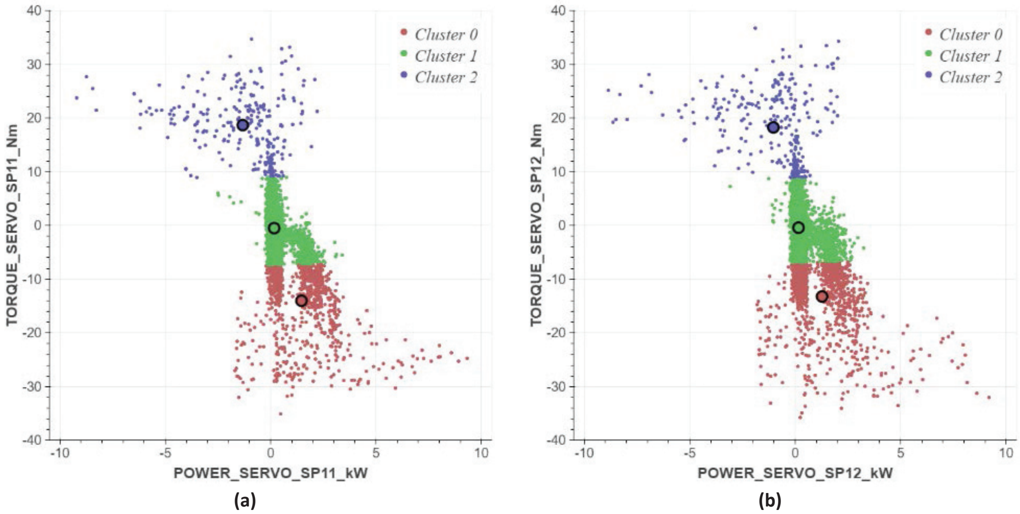

Fig. 3. K-means clustering results (torque vs. power) for (a) Spindle 1 and (b) Spindle 2

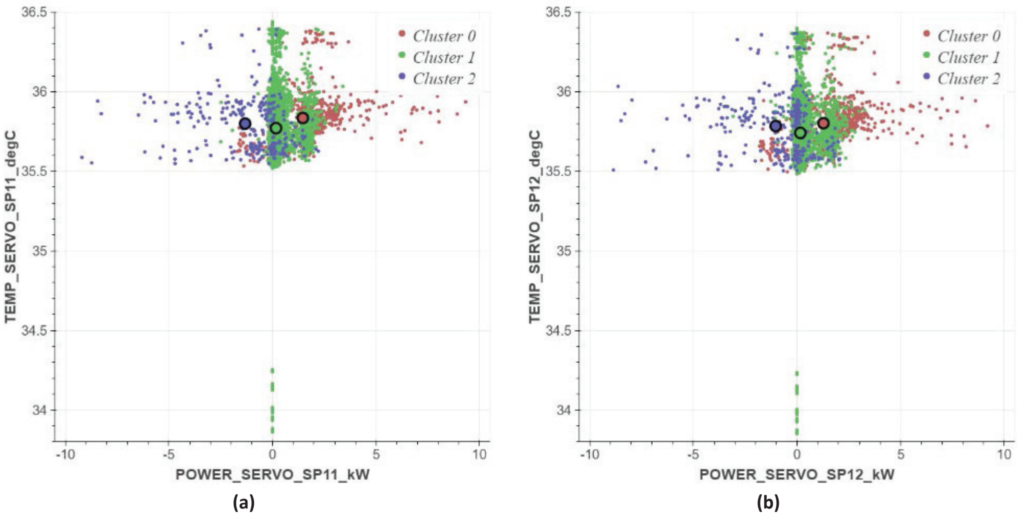

Fig. 4. K-means clustering results (temperature vs. power) for (a) Spindle 1 and (b) Spindle 2

Figure 5 shows spindle torque versus its power consumption as a result of this algorithm. In this case, Cluster 0 ranges are very similar to Cluster 0 found by the k-means algorithm. Cluster 1 is equivalent to Cluster 1 found by the k-means algorithm. Cluster 2 is mainly equivalent to Cluster 2 output by the k-means algorithm. For these dimensions, cluster shapes for each spindle are similar with almost the same number of clustered instances.

The relation between temperature and power is shown in Figure 6. In this case, Cluster 0 and Cluster 2 have a temperature range from 35.5 to 36.5ºC and Cluster 1 from 33.8 to 36.5ºC. In these dimensions, cluster sizes and shapes are similar too. Table 1 summarizes centroid locations in each cluster dimension and the clustered instances per spindle, where it can be seen different centroid coordinates from k-means algorithm.

3.3. Gaussian mixture model clustering

This algorithm has the following parameters: (a) the covariance type, where full type is used, i.e., each cluster has its own unrestricted covariance matrix; (b) initialization, where k-means is used to get initial cluster centroids; and (c) number of clusters k, chosen as described for k-means clustering. The other parameters used were left as Scikit-learn library defaults.

Figure 7 shows spindle torque versus its power consumption as a result of this algorithm. In this case, Cluster 0 is well spread all over the area and is integrated by instances that have a torque between -33 and 33 N·m and a power consumption from -10 to 10 kW. Cluster 1 is integrated by instances that mainly have a torque between -10 and 10 N·m and a very narrow power consumption range from -1 to 1 kW. Cluster 2 is integrated by instances with a torque from -8 to 0 N·m whose power range is from 0 kW to 2 kW, and is completely different from the other algorithms. Across these two dimensions, the shapes and the sizes of each cluster are similar for both spindles with almost the same clustered instances. However, the number of instances per cluster is completely different from agglomerative hierarchical and k-means algorithms.

The relation between temperature and power is shown in Figure 8. In this case, Cluster 2 has a temperature range from 33.8 to 36.5ºC. The range for Cluster 0 and Cluster 1 is from 35.5 to 36.5ºC. Table 1 summarizes centroid locations in each cluster dimension and clustered instances per spindle.

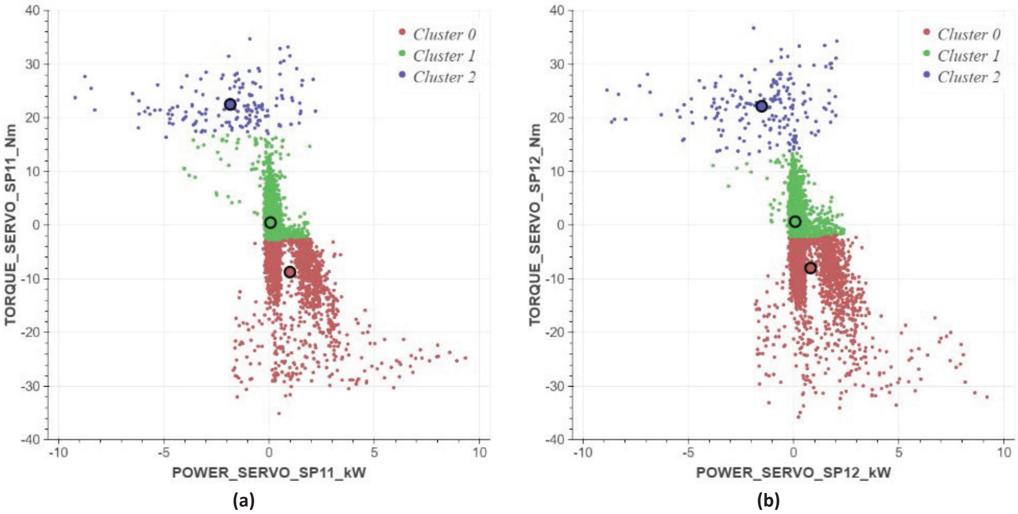

Fig. 5. Agglomerative hierarchical clustering results (torque vs. power) for (a) Spindle 1 and (b) Spindle 2

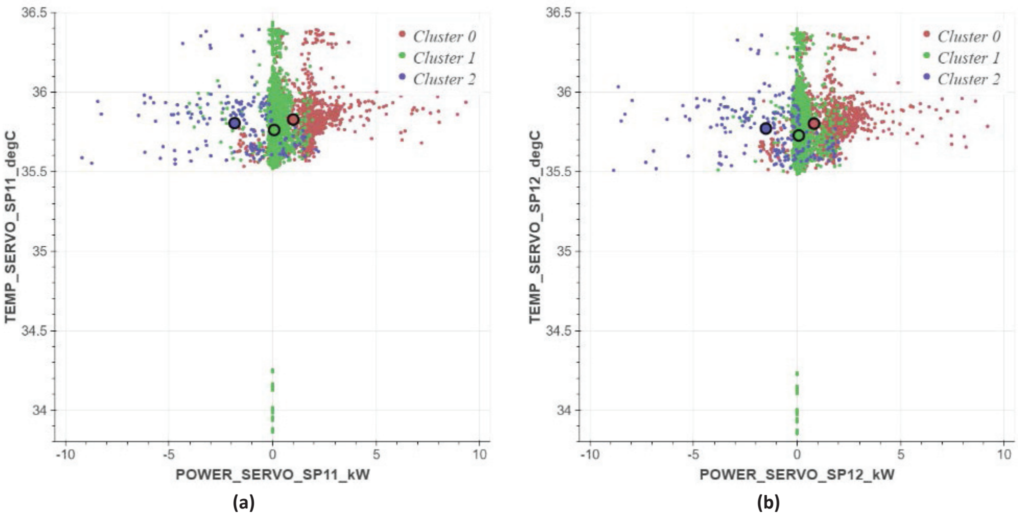

Fig. 6. Agglomerative hierarchical clustering results (temperature vs. power) for (a) Spindle 1 and (b) Spindle 2

4. Discussion

The results can be analyzed from two points of view: clustering algorithms (Section 4.1) and internal spindle behavior (Section 4.2). The purpose of this comparison is to analyze spindle behavior in order to understand how each algorithm describes the machining process and how it can be used for knowledge discovery.

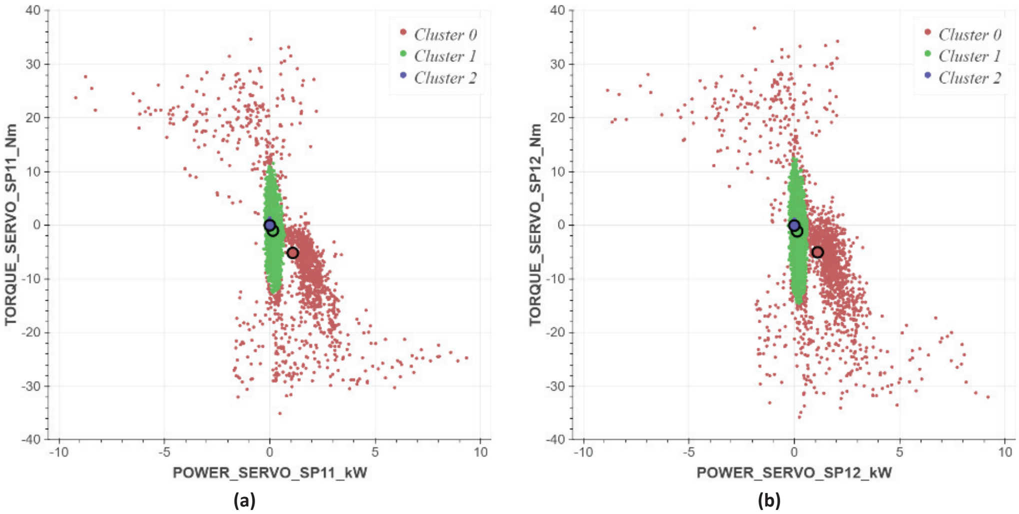

Fig. 7. Gaussian mixture model clustering results (torque vs. power) for (a) Spindle 1 and (b) Spindle 2

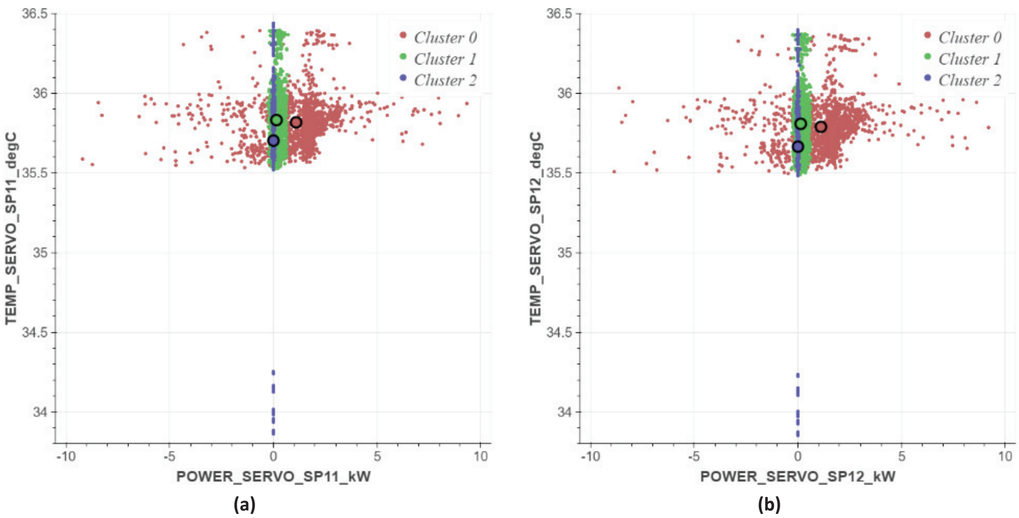

Fig. 8. Gaussian mixture model clustering results (temperature vs. power) for (a) Spindle 1 and (b) Spindle 2

Table 1: Clustering algorithm centroids and number of instances in each cluster for each spindle

		K-means				Agglomerative hierarchical				Gaussian mixture models			
Sp. No.	Cl.	Pow. (kW)	Tor. (N·m)	Tem. (ºC)	No. Instances	Pow. (kW)	Tor. (N·m)	Tem. (ºC)	No. Instances	Pow. (kW)	Tor. (N·m)	Tem. (ºC)	No. Instances
	0	1.4	-14.0	35.8	9,660	1.0	-8.7	35.8	22,145	1.1	-5.1	35.8	15,020
1	1	0.1	-0.5	35.8	79,003	0.1	0.5	35.8	67,537	0.1	-1.0	35.8	38,752
	2	-1.3	19.0	35.8	2,634	-1.8	22.5	35.8	1,615	0.0	0.0	35.7	37,525
	0	1.3	-13.2	35.8	11,518	0.8	-8.0	35.8	26,882	1.1	-5.0	35.8	15,005
2	1	0.1	-0.4	35.7	76,909	0.1	0.6	35.7	62,515	0.1	-1.1	35.8	39,877
	2	-1.0	18.2	35.8	2,870	-1.5	22.0	35.8	1,900	0.0	0.0	35.6	36,415

4.1. Clustering algorithms

From a machine learning point of view, algorithm behavior is very similar. Going into details, however, large amounts of data (more than 90,000 instances) pose a real challenge in terms of processing time for the three algorithms. In order to effectively compare the algorithms according to the same standard, techniques were run on a Linux virtual machine with a 2.2 GHz dual-core processor and 8 GB of RAM.

Accordingly, the slowest was the agglomerative hierarchical algorithm, which took almost 35 seconds to complete the task. It was found to be highly dependent on the connectivity matrix defined in Section 2.2, where it had difficulties finding neighbors, perhaps because of the high dynamics and non-linarites experienced in the process, e.g., when the tool touches the material. As a result, the algorithm had trouble finding clusters to merge. Further experiments using different Python libraries to build the connectivity matrix may perhaps increase overall efficiency. This is, however, beyond the scope of this paper. At the other end of the scale, the fastest algorithm was k-means clustering, requiring fewer than 0.8 seconds to complete the task. This result positions the k-means algorithm as a handy tool for conducting a preliminary exploration of the dataset.

Additionally, the k-means and agglomerative hierarchical algorithms yield similar results. Moreover, high torque cluster (Cluster 2) centroids only differ by around 3.6 kW for both spindles, with the same temperature and almost the same power, meaning that instances in each cluster are common, i.e., Euclidean distances are very similar. Table 2 shows the number of matching instances between clusters found by each algorithm. In this case, only 13,504 and 16,335 instances for Spindle 1 and Spindle 2, respectively, are not shared between clusters, which is around 16%, i.e., 84% of instances are the same for both algorithms. For Spindle 2, the number of shared instances is larger for Cluster 0 and Cluster 2.

However, graphical results for k-means and agglomerative hierarchical clustering differ mainly for Cluster 1, whereas the agglomerative hierarchical algorithm has a wider torque range with fewer clustered instances. We can only make sense of this by analyzing the detailed machining cycle, where toque and power are low, but a positive sign may correspond to tool positioning without any material removal.

Oppositely, the k-means algorithm divides each cluster using the sign of angular speed, not used during clustering but inferred from power and torque, for example, high positive (tool rotating counter-clockwise), high negative (tool rotating clockwise) and low. This does not provide the level of detail found by agglomerative hierarchical clustering, but it might be useful for analyzing spindle behavior at all levels of angular speed: rotating elements may work differently when angular speed is changed.

Conversely, the Gaussian mixture model algorithm works completely differently from the other two. As its clustering technique uses probability distribution mixture model instead of Euclidean distances, it is able to find different points of view. In fact, this is the only algorithm capable of recognizing two main clusters: material removal (Cluster 0) and no material removal (Cluster 1). Cluster 2 corresponds to idle machine, when spindles are stopped.

Table 2: Number of shared instances between k-means and agglomerative hierarchical algorithm results for each spindle

Spindle No.	Cluster	K-means	Agglomerative Hierarchical	Shared instances
	0	9,660	22,145	9,660
1	1	79,003	67,537	66,518
	2	2,634	1,615	1,615
	0	11,518	26,882	11,518
2	1	76,909	62,515	61,534
	2	2,870	1,900	1,900

In this case, the Gaussian mixture algorithm is useful for analyzing the machining process independently of the tool rotating direction and cycle phase. This is useful if we require an overview of machine performance. Note, however, that an increase in the number of clusters will provide more details. In this case, the direction of rotation may be included.

Additionally, the algorithms are spread over the time frame, depending on machining cycles only, formally proven by uniformity $\chi2$ tests, yielding $p > 0.99$ in all cases. This demonstrates that, for this particular case, clustering techniques create time-independent groups.

4.2. Spindle mechanics

From the engineering point of view, the results yielded by the machine learning-based CPS show highly precise elements working in the real world. A comparison of spindle plots gives a clear understanding that both elements are performing the same operation. Clustering techniques capture this situation, showing similar cluster structures independently from the algorithm used.

Therefore, Spindle 2 clusters are more scattered, meaning that this spindle slightly deviated from its optimal operating point. Critical mechanical elements are required to be robust, where repetitiveness is one of the most important indicators: relations between variables should be constant. These relations are difficult to maintain in the real world, as there are other external factors, like noise, that cause deviation. However, we find that the range of torque values for Spindle 2 is larger for the same power value. Nevertheless, this situation does not apply for temperature, as this is a slower variable, which needs some amount of time to change. This phenomenon is known as thermal inertia.

However, the temperature variable shows the effectiveness of cooling systems, which are working to keep spindles between 35.5ºC and 36.5ºC. This range is only 1ºC, which is remarkable in a system with different operating states across each cycle. All three algorithms yielded a mean temperature of around 35.8ºC depending on the cluster. In this way, cooling system behavior can be monitored. Additionally, temperature measurement has other uses, such as an internal malfunction indicator, e.g., a failure in an internal ball-bearing can be detected by an increase in temperature.

Another consideration is that plots show that a small amount of power of no more than 2 kW is used to change rotating direction during machining (Gaussian mixtures models — Cluster 0). High levels of power and torque show that spindles are working at their maximum rated capacity, which means they are correctly selected to perform this operation. Otherwise, the power levels for smaller spindles will be concentrated around high values and for larger spindles at low values.

Regarding cluster density, the behavior of Cluster 2 in GMM is highly interesting because its area is almost insignificant, only visible in Figure 8. However, it has around 37,000 instances closely packed within its boundaries. In terms of the machining process and the relation of this cluster to the idle state, the small dispersion of operating condition data appears to be logical as the spindle is stopped without any external interference due to tool rotation or material removal. Similarly, Cluster 1 with almost the same number of instances is spread over a tiny area defined by clear power, torque and temperature limits. In this case, it also makes sense as the spindle variables are only affected by the tool rotation inertia, i.e., it has to deal with tool and tool holder imbalances only. Conversely, Cluster 0 has a small number of instances spread all over the area. In this case, the cluster shows variable dispersion because of the machining process, in which material and toolpath discontinuities are detected.

For the k-means and agglomerative hierarchical algorithms, this behavior is completely different because these clustering algorithms are governed by the sign of angular speed. In other words, Cluster 1, the largest cluster, is directly related to a stopped spindle, where torque and power levels increase in terms of the need to apply the brakes to hold position. The number of instances is around 77,500 (k-means) and 65,000 (hierarchical) for Cluster 1, meaning that the spindles are stopped for a large percentage of time during the machining cycles. However, Cluster 0 is the second most populated cluster, where a high density area of positive power is yielded near the center of the plot. From the machining point of view, high density areas mean that the behavior of the tool tip is stable during operation after its first contact with material. Cluster 2 is the least populated group, which is directly related to Cluster 1 with a different machining operation phase where the angular speed is the opposite.

5. Conclusions and further work

A general conclusion of this paper is that unsupervised machine learning algorithms embedded in cyber-physical systems are the key enablers for working towards highly precise diagnosis tools. These knowledge discovery applications might be the first step towards in-process diagnosis and then to prognosis tools which would be highly beneficial for new detection-based predictive maintenance applications.

In terms of clustering techniques, the Gaussian mixture model is able to find useful groups more related to operating conditions. Even though it is rather more complex to implement, results were found to be optimal in terms of interpretation by machine tool experts. The agglomerative hierarchical algorithm is useful for analyzing the cycle's phases in detail. However, it is highly influenced by inherent variables like angular speed. This may interfere with the interpretation or new knowledge discovery. K-means clustering has the same characteristics as the agglomerative hierarchical algorithm in terms of inherent variables. However, it is appreciably faster, being a powerful tool for rapidly analyzing datasets.

In terms of algorithm developments, we intend to adapt a static clustering analysis to a mainly dynamic process in future research. In this way, knowledge discovery will be able to run in an online manner, helping to understand cluster evolution in terms of cluster shapes (how the identified machining characteristics change over time) and number of clusters (identify new machining characteristics). With a view to real-time operation, however, further research in this field should focus on upgrading CPS embedded electronics to enable the algorithm implementation on its FPGA.

Acknowledgements

This research has been supported by the Spanish Centre for the Development of Industrial Technology (CDTI) through the TIC-20150093 project and partially supported by the Spanish Ministry of Economy and Competitiveness through the TIN2016-79684-P project and by the Regional Government of Madrid through the S2013/ICE-2845-CASI-CAM-CM project. We would like to thank Etxe-Tar, Ikergune and SoC-e for providing experimental setup.

References

[1] Baheti, R., & Gill, H., 2011. Cyber-physical systems. *The Impact of Control Technology*, 12, 161-166.

[2] Lee, J., Kao, H. A., & Yang, S., 2014. Service innovation and smart analytics for Industry 4.0 and big data environment. *Procedia CIRP*, 16, 3-8.

[3] Pandit, S. M., Paul, D., & Roth, J. T., 1996. Monitoring and diagnosis of bearing defects using data dependent systems. *Integrated Computer-Aided Engineering*, 3(4), 268-278.

[4] Wang, Y., Peng, Y., Zi, Y., Jin, X., & Tsui, K. L., 2016. A two-stage data-driven-based prognostic approach for bearing degradation problem. *IEEE Transactions on Industrial Informatics*, 12(3), 924-932.

[5] Tobon-Mejia, D. A., Medjaher, K., & Zerhouni, N., 2012. CNC machine tool's wear diagnostic and prognostic by using dynamic Bayesian networks. *Mechanical Systems and Signal Processing*, 28, 167-182.

[6] Yan, J., Koc, M., & Lee, J., 2004. A prognostic algorithm for machine performance assessment and its application. *Production Planning & Control*, 15(8), 796-801.

[7] Choudhary, A. K., Harding, J. A., & Tiwari, M. K., 2009. Data mining in manufacturing: a review based on the kind of knowledge. *Journal of Intelligent Manufacturing*, 20(5), 501-521.

[8] Gardner, M., & Bieker, J., 2000. Data mining solves tough semiconductor manufacturing problems. In Proceedings *of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 376-383). ACM.

[9] Chien, C. F., Wang, W. C., & Cheng, J. C., 2007. Data mining for yield enhancement in semiconductor manufacturing and an empirical study. *Expert Systems with Applications*, 33(1), 192-198.

[10] Sebzalli, Y. M., & Wang, X. Z., 2001. Knowledge discovery from process operational data using PCA and fuzzy clustering. *Engineering Applications of Artificial Intelligence*, 14(5), 607-616.

[11] Liao, T. W., Li, D. M., & Li, Y. M., 1999. Detection of welding flaws from radiographic images with fuzzy clustering methods. *Fuzzy sets and Systems*, 108(2), 145-158.

[12] Jin, Y., & Ishino, Y., 2006. DAKA: design activity knowledge acquisition through data-mining. *International Journal of Production Research*, 44(14), 2813-2837.

[13] Piatetsky-Shapiro, G., 1996. *Advances in Knowledge Discovery and Data Mining* (Vol. 21). U. M. Fayyad, P. Smyth, & R. Uthurusamy (Eds.). Menlo Park: AAAI press.

[14] Hastie, T., Tibshirani, R., & Friedman, J. (2009). Unsupervised learning. In *The Elements of Statistical Learning* (pp. 485-585). Springer New York.

[15] Nandi, S., Toliyat, H. A., & Li, X., 2005. Condition monitoring and fault diagnosis of electrical motors-a review. *IEEE Transactions on Energy Conversion*, 20(4), 719-729.

[16] Witten, I. H., & Frank, E., 2005. *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann.

[17] Jain, A. K., 2010. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8), 651-666.

[18] Ward Jr, J. H., 1963. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301), 236-244.

[19] McLachlan, G., & Peel, D., 2004. *Finite Mixture Models*. John Wiley & Sons.

[20] Pedregosa, F., Varoquaux, G., et al., 2011. Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.

[21] Milligan, G. W., & Cooper, M. C., 1985. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2), 159-179.

[22] Arthur, D., & Vassilvitskii, S., 2007. K-means++: The advantages of careful seeding. *In Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (pp. 1027-1035). Society for Industrial and Applied Mathematics.

[23] Xu, D., & Tian, Y., 2015. A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2), 165-193.