Journal of Machine Engineering, 2023, Vol. 23, No. 4, 5–17
ISSN 1895-7595 (Print) ISSN 2391-8071 (Online)

Received: *13 June 2023* / Accepted: *21 August 2023* / Published online: *25 August 2023*

feature extraction, k-means clustering,
times series,
unsupervised learning

Fabio C. ZEGARRA[1], Juan VARGAS-MACHUCA[1],
Avid ROMAN-GONZALEZ[2], Alberto M. CORONADO[1*]

AN APPLICATION OF MACHINE LEARNING METHODS TO CUTTING TOOL PATH CLUSTERING AND RUL ESTIMATION IN MACHINING

Machine learning has been widely used in manufacturing, leading to significant advances in diverse problems, including the prediction of wear and remaining useful life (RUL) of machine tools. However, the data used in many cases correspond to simple and stable processes that differ from practical applications. In this work, a novel dataset consisting of eight cutting tools with complex tool paths is used. The time series of the tool paths, corresponding to the three-dimensional position of the cutting tool, are grouped according to their shape. Three unsupervised clustering techniques are applied, resulting in the identification of DBA-*k*-means as the most appropriate technique for this case. The clustering process helps to identify training and testing data with similar tool paths, which is then applied to build a simple two-feature prediction model with the same level of precision for RUL prediction as a more complex four-feature prediction model. This work demonstrates that by properly selecting the methodology and number of clusters, tool paths can be effectively classified, which can later be used in prediction problems in more complex settings.

1. INTRODUCTION

The production of goods has been transformed by the convergence of various technological advances, both physical and digital. This convergence, made possible by a combination of technologies such as new sensors, automation, machine learning, and cloud computing, is also known as Industry 4.0 or the fourth industrial revolution [1]. Although Industry 4.0 holds great promise, its real-world applications on production lines are limited due to difficulties in interoperability between different production entities and the lack of a control system adapted to the expected flexibility and management of the generated data. Addressing these diverse problems is crucial to realizing the full potential of Industry 4.0 [2, 3].

[1] Laboratorio de Sistemas Inteligentes, EPIME, Universidad Nacional Tecnológica de Lima Sur (UNTELS), Peru
[2] Aerospace Sciences & Health Research Laboratory (INCAS-Lab), Universidad Nacional Tecnológica de Lima Sur (UNTELS), Peru
[*] E-mail: acoronado@untels.edu.pe
https://doi.org/10.36897/jme/171432

Manufacturing execution systems (MESs) are computerized systems that support manufacturing execution processes from production order release until the delivery of finished goods [4]. MESs are increasingly important in the digital transformation era of Industry 4.0, where high levels of digital capabilities play an essential role in the factories of the future. MESs aid businesses in achieving their Industry 4.0 goals through increased operation visibility and traceability. Combining MES with digital twins provides simulation capabilities to trace, analyse, and optimize a production process using a virtual replica [5]. Vyskočil et al. [6] propose an architecture for an MES that can autonomously create and execute production plans using digital twins and symbolic planning methods. The proposed solution allows for efficient production by starting with an initial plan and then continuously searching for better alternatives while the plan is being executed.

Advances in sensor technology have enabled a wide range of new applications, such as indirect condition monitoring of machine tools [7]. Researchers have published studies using data from a variety of sensors, including force, vibrations, acoustic emission, temperature, sound, electric current, and light [8]. The measurements from these sensors are typically time series data, which can require a large amount of storage space depending on the specific application. In the early days of Industry 4.0, data was rarely analysed. Now, with more data sources, there is a growing awareness of the importance of extracting information from data. This can create value by facilitating data reduction to actionable items [9].

Machine learning is one of the most promising technologies for manufacturing, especially in machining processes. Applications of data analytics in various machining processes have increased significantly since the 1980s, with a major emphasis after 2015. There are currently many applications, such as monitoring chatter, roughness, quality, and especially condition [10]. Machine learning algorithms can use the time series mentioned above directly or after preprocessing. A wide range of algorithms have been shown to be useful in condition monitoring problems, including artificial neural networks, k-means, support vector machines, decision trees (and their variations), among others [11]. Depending on whether the variable to be predicted is available, some of these algorithms can be applied in a supervised or unsupervised manner.

Machine tool components can exhibit a variety of faults, including excessive friction, loose motor, wrong commutation offset, and wear. These faulty states can be recreated during normal operation or artificially induced. A diverse set of measurements, such as position, force, and vibration, can be used to identify the condition of machine tool components using unsupervised algorithms. These algorithms create clusters that can segregate the different faults. For this task, most algorithms perform reasonably well when using a combination of statistical and spectral feature extraction methods [12].

Predicting the remaining useful life (RUL) of machine tools under real-world conditions poses a significant challenge, particularly when faced with dynamic operating conditions such as widely fluctuating spindle loads and intricate tool paths. Researchers have employed a range of methods, from traditional machine learning techniques like support vector machines to deep learning approaches with attention mechanisms, to tackle more complex cutting shapes [13, 14]. The reported results show scores ranging from 0.2 to 0.8 on a scale of 0 to 1, which are relatively less precise than those obtained for machine tool wear problems, such as the widely known PHM 2010 dataset collected in laboratory settings [14].

This article has five sections. The first section reviews the current trends in manufacturing, such as Industry 4.0 and manufacturing execution systems (MESs). It also explores the diverse applications of machine learning for condition monitoring of machine tools. It discusses unsupervised and supervised methods, especially in the context of remaining useful life (RUL) and wear prediction. The second section explains the unsupervised and supervised machine learning methods that this study uses. The third section describes the experimental setup for data collection, which corresponds to the 2020 Foxconn Industrial AI Data Challenge, and the procedure that this work proposes. The fourth section analyses the results and discusses their implications. The fifth section concludes the article.

2. MACHINE LEARNING METHODS

Machine learning methods use algorithms to learn from data and then make predictions. There are two main types of machine learning methods: unsupervised and supervised. Unsupervised methods cluster (group) unlabeled data, without human guidance. They discover hidden patterns and structures in data that humans might miss. Supervised methods train or "supervise" algorithms with labelled data, which have predefined inputs and outputs. They classify data into categories (for example: no wear, medium wear, severe wear) or predict outcomes based on data relationships (for example: minutes until breakdown or tool wear in mm).

2.1. STANDARD EUCLIDEAN K-MEANS

The *k*-means is a widely used unsupervised classification method. It uses an algorithm that divides data into *k* groups and minimizes the sum of squares within each group. This algorithm creates clusters from data. A kernel function can enhance the *k*-means algorithm by applying a nonlinear mapping from the original space to a higher dimensional feature space. This mapping enables the extraction of groups that are not linearly separable in the input space [15]. The *k*-means algorithm assigns a set of observations, represented as feature vectors, to groups based on their similarity. This similarity depends on a distance metric of *k* centroids, where the centroid is the center of a group based on the mean of its members. In the standard case, the algorithm uses Euclidean distance as the similarity metric.

2.2. DBA-K-MEANS

Another unsupervised method is dynamic time warping (DTW), which originated from applications in speech recognition and relies on the Levenshtein distance [16]. This method finds the optimal alignment between two sequences of numerical values and captures flexible similarities by aligning coordinates within both sequences. DTW barycenter averaging (DBA) uses a heuristic strategy as a global average method. DBA iteratively refines an initial average sequence to minimize its squared distance (DTW) to the averaged sequences [17].

To apply the *k*-means clustering algorithm to time series with dynamic time warping, two modifications are needed. First, the algorithm should use Dynamic Time Warping (DTW) as a distance measure to group time series of similar shapes. Second, the algorithm should compute cluster centroids (or barycenters) with respect to DTW. A barycenter is the mean sequence of a group of time series in DTW space. The DTW Barycenter Averaging (DBA) algorithm can compute the barycenter by minimizing the sum of the squared DTW distance between the barycenter and the series in the group. Alternatively, the smooth DTW algorithm can compute the barycenter by minimizing the weighted sum of the smooth DTW distances between the barycenter and the series in the group. Researchers can adjust these weights, but they must add up to 1.

2.3. AUTOENCODER–LSTM-*k*-means

An additional example of an unsupervised method is the autoencoder, a type of neural network model that learns to generate its input as output. The encoder first compresses the input data into a compact representation, also known as a bottleneck. The decoder then reconstructs the input data from the encoded features [18]. By minimizing the reconstruction error during training, the autoencoder learns low-dimensional features with essential attributes of the input data. An LSTM network can be used to implement an autoencoder and model long-term dependencies [19]. After obtaining the compressed representation, it can be used with the *k-means* algorithm for clustering.

2.4. SUPPORT VECTOR REGRESSION (SVR)

Support vector machines (SVM) is a well known supervised machine learning algorithm for classification and regression [20]. After training, support vector regression (SVR) estimates a function that maps an input data to a real number. Like support vector classification (SVC), SVR also maximizes the margin and uses the kernel trick for nonlinear mapping. SVR is a non-parametric technique because it relies on kernel functions. All kernel functions must meet the Mercer condition that corresponds to the inner product of some feature space [21]. Some common kernels are: linear, polynomial, sigmoid and radial basis function (RBF).

3. DATA DESCRIPTION AND PROCEDURE

3.1. DATA DESCRIPTION

The 2020 Foxconn Industrial AI Data Challenge dataset provides the data for this study, which estimates the remaining useful life (RUL) of cutting tools. The controller (PLC) collects force and spatial position signals at 33 Hz, as Fig. 1 shows. Moreover, add-on sensors measure three-axis vibrations and electric current signals at 25600 Hz.

The challenge organizers give training data from three cutting tools for their whole lifespan. For testing, they give only partial data from five cutting tools. They provide only about one minute of measurements for every five minutes of experiments, due to the large amount of data generated.

Figures 3 and 4 show that the tool paths of the machining tools in this dataset are more complex than those usually reported in the literature. This complexity makes the RUL prediction problem very challenging, as the scores of the data challenge participants show. The scores are from 0 to 100 points, but only the first four teams score above 80.

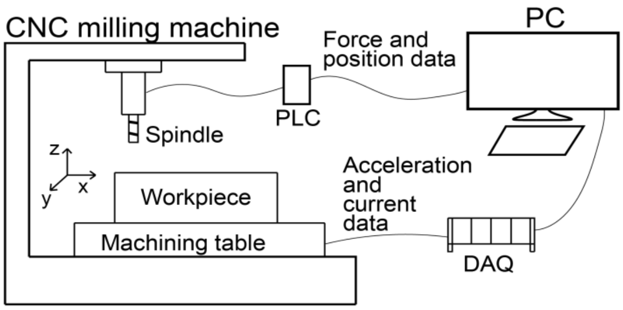

Fig. 1. Diagram of the experimental setup

3.2. PROCEDURE

The unsupervised part of this study follows the flowchart in Fig. 2. The dataset includes training data (01, 02 and 03) and testing data (01, 02, 03, 04 and 05). The procedure combines all the data for unsupervised training.

The x, y, and z position signals represent the tool path in the training and testing data. The time series lengths vary in each csv file, so the procedure reduces them to the smallest length. Then it scales the time series between 0 and 1.

The procedure joins the time series for unsupervised clustering. It uses three methods: autoencoder-LSTM-*k*-means, Euclidean *k*-means, and DBA-*k*-means. The autoencoder-LSTM-*k*-means method compresses the encoder data for the *k*-means algorithm. The LSTM architecture has a layer with 128 units and uses the ReLU activation function. The Euclidean *k*-means and DBA-*k*-means methods use the *tslearn* Python library [22]. For DBA-k-means, the model selects two values for the centroid seeds and uses 100 iterations to estimate the barycenter.

The supervised part of this study follows a similar procedure as Zegarra el al. proposed (see Fig. 10 in [14]). This procedure consists of three main steps: time series preprocessing, feature extraction, and SVR modeling. In the first step, the data undergoes filtering and detrending. In the second step, the following features are extracted: RMS, variance, maximum, skewness, kurtosis, spectral skewness and kurtosis, wavelet energy. In the third step, the data is divided into 70% for training and 30% for testing, and SVR models are trained with the help of Bayesian optimization of hyperparameters.

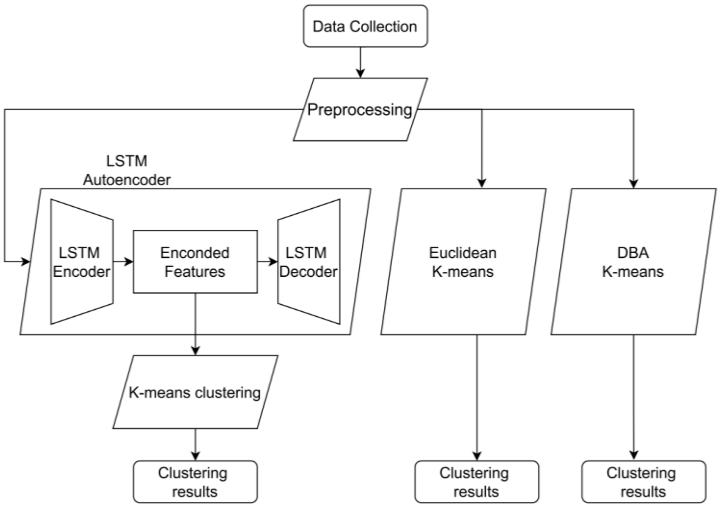

Fig. 2. Flowchart of the procedure

4. RESULTS AND DISCUSSION

4.1. DATASET EXPLORATION

Figure 3 presents selected tool path slices from the datasets (originally in CSV format) for the three training cutting tools under study. The analysis excluded CSV files containing corrupted data from consideration. The tool paths are positioned at the start, middle, and end of each dataset. Note that the specific training data influences the tool paths' shape. For example, train 1 follows a path from the upper-left portion of the workpiece to the lower-right section, moving downwards as time goes by (as seen in the top row of Fig. 3). In contrast, train 2 traverses the periphery of the component from top to bottom, leaving behind vertical lines indicative of the idling movements of the cutting tool (displayed in the central row of Fig. 3). Lastly, train 3 employs a combination of tool paths resembling those of trains 1 and 2, with a mirrored 'T' shape situated in the middle of the working piece (illustrated in the bottom row of Fig. 3). The test data also exhibits diverse tool paths, adding complexity to the challenge of predicting the Remaining Useful Life (RUL).

Figure 4 shows the top view of the tool paths for the eight cutting tools in their entirety. The tool paths are quite dissimilar, and it would be very convenient to have a method that allows accurate and automatic path classification. Some apparent similarity can be found in the tool paths corresponding to train 1, the right side of train 3, test 4, and test 5. However, the remaining tool paths have very dissimilar shapes. The figure also shows that the cutting tool idles several times during the process (diagonal displacements).

The size of measurements in manufacturing processes varies significantly, resulting in time series of different lengths and qualities. The data was originally stored in CSV files, each of which corresponded to approximately one minute of measurements. Figure 5 shows that the CSV files have different lengths and contain corrupted data and other irregularities. These issues must be located and eliminated before further analysis can be performed.

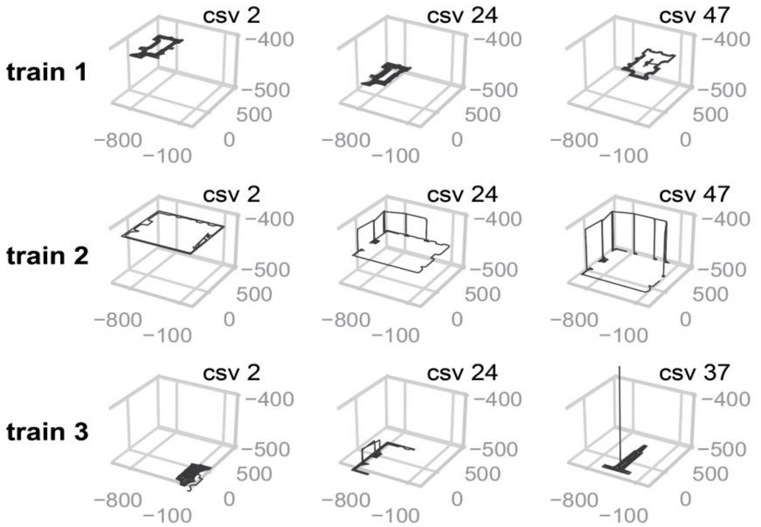

Fig. 3. Tool paths for selected CSV files located at the beginning, middle, and end of the train dataset

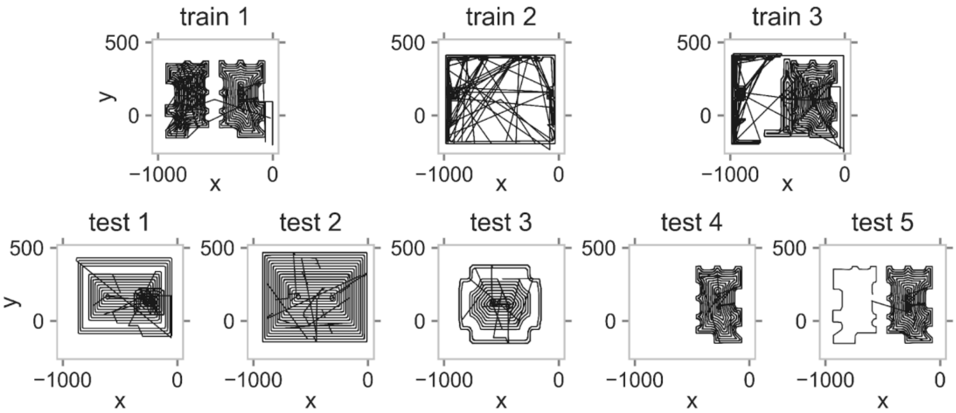

Fig. 4. Tool paths for the whole dataset

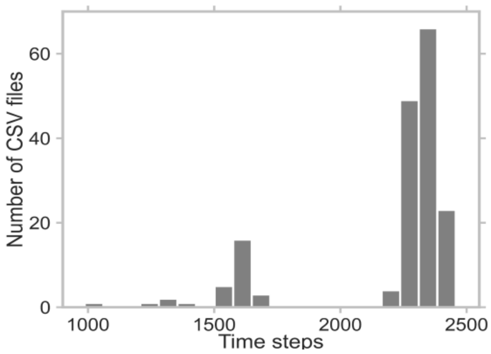

Fig. 5. Histogram for the size in time steps of the CSV files for the whole dataset

4.2. UNSUPERVISED CLASSIFICATION OF TOOL PATHS

Three methods for unsupervised classification of tool paths using three clusters are compared in Figure 6: DBA-k-means (green), standard Euclidean k-means (red), and LSTM-autoencoder-*k*-means (blue). The figure shows the results for the first few CSV files of each cutting tool. The database includes three cutting tools labeled as "train" and five cutting tools labeled as "test". The "train" tools were measured from the beginning to the end of their useful life, while the "test" tools were measured only in a part of their useful life. Each CSV file has an approximate duration of 1 minute and was measured every 5 minutes. Corrupted CSV files, such as the first file of the three train tools, were removed. The figure reveals that the LSTM-autoencoder-*k*-means method differs the most from the other two methods (21 times), assigning several CSV files to cluster 2. The standard Euclidean *k*-means and DBA-*k*-means methods have the lowest number of differences with respect to the other two methods (2 and 1 times, respectively). Therefore, only DBA-*k*-means will be used for further analysis.

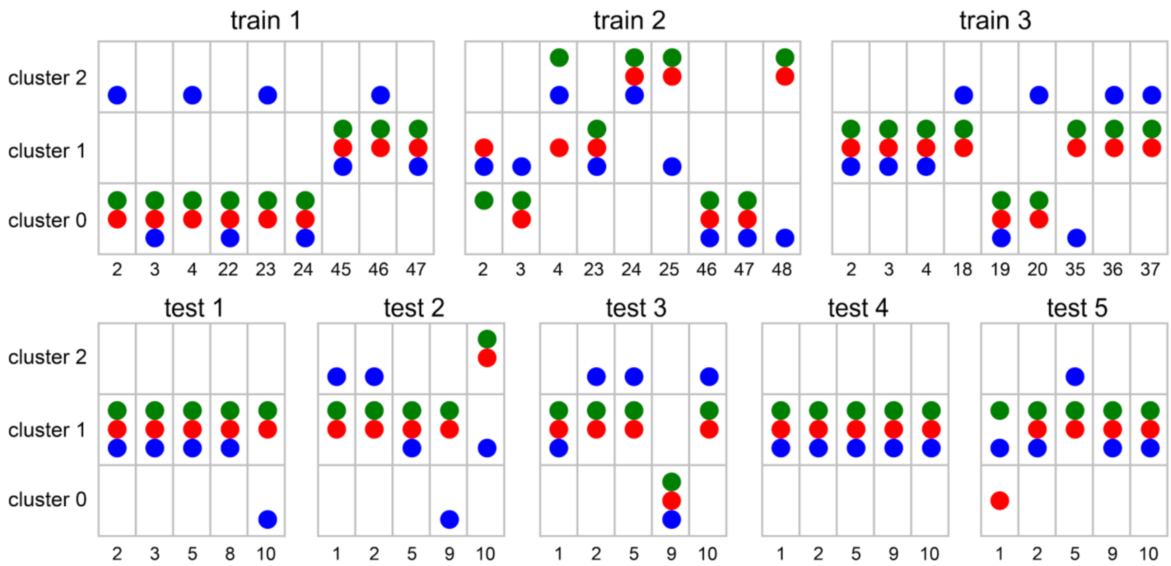

Fig 6. Three unsupervised classification methods, using three clusters: DBA-*k*-means (green), standard Euclidean *k*-means (red), LSTM-autoencoder-*k*-means (blue)

A detailed analysis of the impact of the number of clusters on the way in which the tool paths are grouped appears in Fig. 7. Figure 7a shows the grouped tool paths considering only two clusters. Cluster 0 (left) tends to agglutinate tool paths on the left side, while cluster 1 (right) agglutinates tool paths on the right side. These elements predominate in the tools train 1 (upper left) and train 3 (upper right) of the machine tools (Fig. 4). The tool paths corresponding to train 2 (upper middle) divide equally between both clusters, which is not the most appropriate solution.

Figure 7b illustrates the case of using three clusters. Clusters 0 (left) and 1 (middle) remain similar to the previous case. However, an additional cluster 2 (right) now incorporates elements related to the train 2 tool (Fig. 4). Despite this, the separation is not complete, which could account for a significant portion of the divergent results displayed in Fig. 6. Figure 7c demonstrates that using four clusters does not improve the separation of tool paths. Only when five clusters are employed (Fig. 7d) do clearly separated and defined tool paths emerge. Although a few incorrectly classified CSV files persist, using five clusters seems more appropriate in this situation. The elbow method is a way to find the optimal number of clusters for an unsupervised classification algorithm, such as k-means. The elbow method also recommends five clusters as the optimal number.

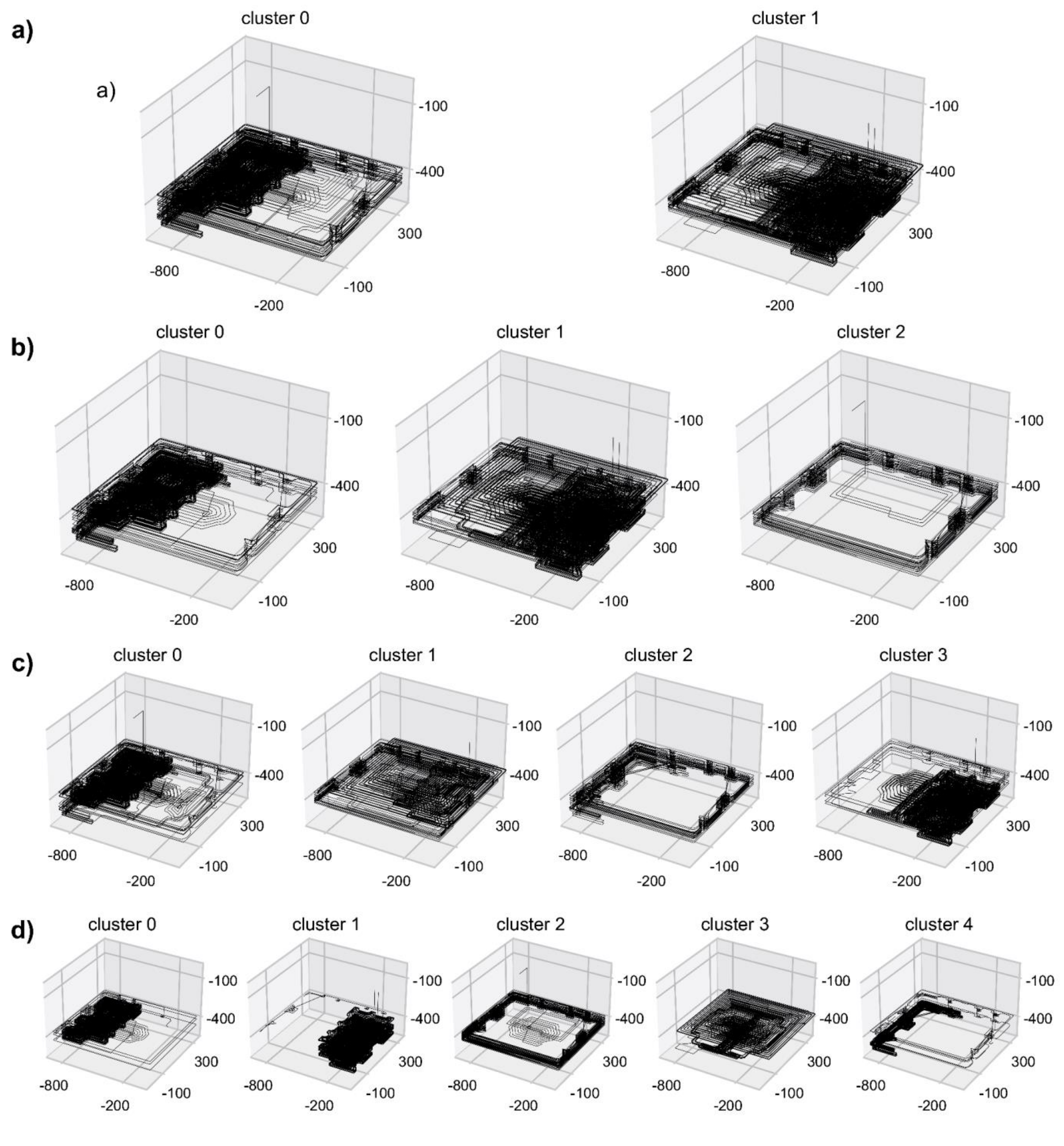

Fig. 7. Tool paths clusters generated by DBA-*k*-means unsupervised classification considering a) two clusters, b) three clusters, c) four clusters and d) five clusters

The DBA-*k*-means method produces the unsupervised classification results in Fig. 8, which shows cases with two, three, four, and five clusters. The tool paths of the train 2 tool in Fig. 4 differ significantly from all other tools, except maybe for the train 3 tool (left side). In Fig. 8, two clusters do not reveal the distinct tool path of train 2. Three clusters make train 2 stand out from the other tools, but cluster 1 almost entirely contains the five test tools, which may not be accurate, as Fig. 4 suggests. Four clusters blur the unique tool path of train 2, as it shares some shapes with other tools. Five clusters indicate that the first three test tools are somewhat related (through cluster 3), as are the last two test tools (through cluster 1).

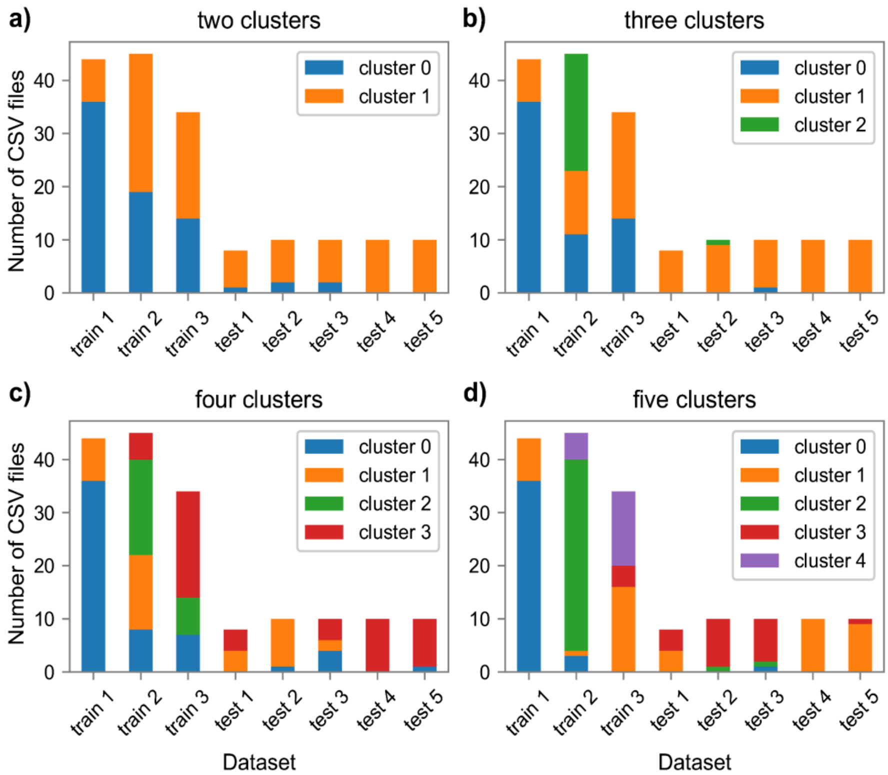

Fig. 8. Contribution of each cluster (from 0 to 4) to the tool path observed on each machine tool (train 1-3 and test 1-5), considering two to five clusters

Figure 8d reveals a significant challenge for developing remaining useful life (RUL) prediction models using data from tools in train 1, 2, and 3 and testing them with data from tools in test 1, 2, 3, 4, and 5. To address this issue, it is advisable to ensure that the training tool paths resemble those in validation and testing. However, in this specific case, many tool paths in train 1 to 3 lack equivalents in the tool paths observed in test 1 to 5.

Despite the difficulties associated with this dataset, studying configurations like this one is crucial for advancing the development of better prediction models. Relying solely on data derived from simple and similar tool paths does not significantly contribute to the challenge of implementing these systems in situations that mirror the complexity found in the manufacturing industry.

4.3. SUPERVISED REGRESSION OF REMAINING USEFUL LIFE (RUL)

In Fig. 8d, for the case of five clusters, it can be observed that cluster 1 is partially present in train 1 and train 3. In addition, this cluster appears almost completely in both test 4 and test 5. Thus, in the following analysis, train 1 and train 3 are used as training data to predict the RUL of test 1, 2, 3, 4 and 5. The detailed methodology to perform the prediction task can be found in previous works [14], [23]. As an initial step, a search of features was carried out using the forward feature engineering method on the training data (train 1 and train 3). In this search, two features (hence the name *set-2f*): Dvib1_mean_abs_change and *Dz*_maximum, were selected. Then, 500 RUL predictions were made, which are presented as boxplots for each test in Fig. 9.

Figure 9 shows a comparison of the boxplots of the prediction of the RUL for two test cases. The first one, named *set-4f*, corresponds to the case of using train 1, train 2 and train 3 as training data [14], [23]. In this case, it was necessary to search for up to four features (hence the name *set-4f*): Dvib1_mean_abs_change, loadF_absolute_sum_of_changes, Dvib1_abs_energy and Dvib3_mean_abs_change, to obtain results reasonably precise (score above 0.8). As a reference, the actual RULs for test 1, test 2, test 3, test 4 and test 5 are 104, 52, 190, 66 and 40 min, respectively. These actual RULs are shown as red dashed lines in Fig. 9.

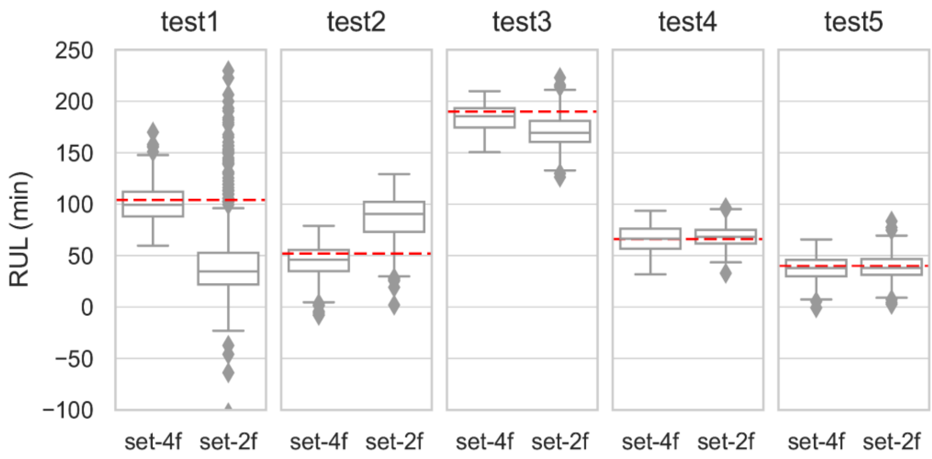

Fig. 9. Boxplots comparing the remaining useful life (RUL) predictions for the set-*4f* and set-*2f* test cases. The red dashed lines indicate the actual RUL for each test

The results shown in Fig. 9 demonstrate that by properly selecting the training data (in this case, train 1 and train 3) and using only two features (*set-2f*), it is possible to adequately predict the RUL of test 4 and test 5. These results are comparable to much more complex models (*set-4f*) that use more data and take more time to develop. These results validate the recommendations obtained when analysing the results of the unsupervised model, which, considering five clusters, indicated that the tool paths of train 1 and train 3 presented a higher similarity to the tool paths of test 4 and test 5, facilitating the construction of predictive models in a simpler and faster way.

5. CONCLUSIONS

The present study examines a dataset of eight machine tools with significant variations in their tool paths, posing challenges for accurate classification and regression models. The train 1 tool exhibits well-defined tool paths, while train 2 proves to be the most difficult to classify accurately. The use of DBA-*k*-means and standard Euclidean *k*-means methods yields more consistent results than LSTM-autoencoder-k-means. Additionally, the number of clusters greatly influences the grouping of tool paths, with five clusters providing clearer separation for this particular dataset. The results from the unsupervised classification demonstrate that cluster 1 dominates test 4 and test 5 cutting tools and is also present in train 1 and train 3 cutting tools. After using train 1 and train 3 as training data and test 4 and test 5 as testing data, it is possible to observe that selecting only two features yields similar results as using a more complex model that considers four features and more training data.

This work emphasizes the importance of studying machining processes with complex tool paths to advance the development of more robust prediction models with real-world applications. Machine learning and data analytics significantly impact many industries, including manufacturing, in several ways. In the broad field of manufacturing, particularly in machining processes, various contexts have benefited from data analysis, including the prediction of machine tool remaining useful life (RUL). Notably, past works have considered relatively simple and stable cutting processes, which can greatly differ from operating conditions in practical applications. This article presents a methodology for directly analysing machining processes to better group similar cutting tools according to their tool paths. These tool paths are expected to be even more complex in industrial processes; hence, a method-logy such as this one becomes even more relevant. Properly classifying cutting tools will not only allow automatic grouping of experimental data but also help determine if there is sufficient and appropriate data to solve problems such as wear and remaining useful life (RUL) prediction, among others.

ACKNOWLEDGEMENTS

This work was funded by *CONCYTEC-FONDECYT in the framework of call E038-01, grant No. 020-2019-FONDECYT-BM-INC.INV.*

REFERENCES

[1] RAI R., TIWARI M.K., IVANOV D., DOLGUI A., 2021, *Machine Learning in Manufacturing and Industry* 4.0 *Applications*, Int. J. Prod. Res., 59, 4773–4778.
[2] MAYNARD A.D., 2015, *Navigating the Fourth Industrial Revolution*, Nat. Nanotechnol., 10/12, 1005–1006.
[3] TETI R., MOURTZIS D., ADDONA D.M.D., CAGGIANO A., 2022, *Process Monitoring of Machining*, CIRP Annals, 71/2, 529–552.
[4] SAENZ de UGARTE B., ARTIBA A., PELLERIN R., 2009, *Manufacturing Execution System – a Literature Review*, Prod. Plan. Control, 20, 525–539.
[5] BENFRIHA K., et al., 2021, *Development of an Advanced MES for the Simulation and Optimization of Industry 4.0 Process*, Int. J. Simul. Multi. Design Optim., 12, 23.
[6] VYSKOCIL J., DOUDA P., NOVAK P., WALLY B., *2023*, *A Digital Twin-Based Distributed Manufacturing Execution System for Industry 4.0 with AI-Powered on-the-Fly Replanning Capabilities*, Sustain. Sci. Pract. Policy, 15/7, 6251.
[7] TETI R., JEMIELNIAK K., O'DONNELL G., DORNFELD D., 2010, *Advanced Monitoring of Machining Operations*, CIRP Annals, 59/2, 717–739.
[8] KUNTOGLU M., et al., 2020, *A Review of Indirect Tool Condition Monitoring Systems and Decision-Making Methods in Turning: Critical Analysis and Trends*, Sensors, 21/1, https://doi.org/10.3390/s21010108.
[9] RAPTIS T.P., PASSARELLA A., CONTI M., 2019, *Data Management in Industry 4.0: State of the art and Open Challenges*, IEEE Access, 7, 97052–97093.

[10] AGGOGERI F., PELLEGRINI N., TAGLIANI F.L., 2021, *Recent Advances on Machine Learning Applications in Machining Processes*, Applied Sciences, https://doi.org/10.3390/app11188764.
[11] MOHAMED A., HASSAN M., M'SAOUBI R., ATTIA H., 2022, *Tool Condition Monitoring for High-Performance Machining Systems–A Review*, Sensors, 22/6. 2206, https://doi.org/10.3390/s22062206.
[12] GITTLER T., SCHOLZE S., RUPENYAN A., WEGENER K., 2020, *Machine Tool Component Health Identification with Unsupervised Learning*, Journal of Manufacturing and Materials Processing, 4/3, 86, https://doi.org/10.3390/jmmp4030086.
[13] LIU C., ZHANG L., NIU J., YAO R., WU C., 2020, *Intelligent Prognostics of Machining Tools Based on Adaptive Variational Mode Decomposition and Deep Learning Method with Attention Mechanism*, Neurocomputing, 417, 239–254.
[14] ZEGARRA F.C., VARGAS-MACHUCA J., CORONADO A.M., 2021, *Tool Wear and Remaining Useful Life (RUL) Prediction Based on Reduced Feature Set and Bayesian Hyperparameter Optimization*, Prod. Eng., 16/4, 465–480.
[15] DHILLON I.S., GUAN Y., KULIS B., 2004, *Kernel K-Means*, Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining – KDD, https://doi.org/10.1145/1014052.1014118.
[16] SAKOE H., CHIBA S., 1978, *Dynamic Programming Algorithm Optimization for Spoken Word Recognition*, IEEE Transactions on Acoustics, Speech, and Signal Processing, 26/1. 43–49, https://doi.org/10.1109/tassp.1978.1163055.
[17] PETITJEAN F., KETTERLIN A., GANÇARSKI P., 2011, *A Global Averaging Method for Dynamic Time Warping, with Applications to Clustering*, Pattern Recognition, 44/3. 678–693, https://doi.org/10.1016/j.patcog.2010.09.013.
[18] GOODFELLOW I., BENGIO Y., COURVILLE A., 2016, *Deep Learning*, MIT Press.
[19] REN W., WEN G., ZHANG Z., MAZUMDER J., 2021, *Quality Monitoring in Additive Manufacturing Using Emission Spectroscopy and Unsupervised Deep Learning*, Materials and Manufacturing Processes, 1–8, https://doi.org/10.1080/10426914.2021.1906891.
[20] VAPNIK V.N., 2000, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York.
[21] VAPNIK V.N., 1999, *An Overview of Statistical Learning Theory*, IEEE Trans. Neural Netw., 10/5, 988–999.
[22] TAVENARD R., et al., 2020, *Tslearn, A Machine Learning Toolkit for Time Series Data*, J. Mach. Learn. Res., 21, 118, 1–6.
[23] CASUSOL A.J., ZEGARRA F.C., VARGAS-MACHUCA J., CORONADO A.M., 2021, *Optimal Window Size for the Extraction of Features for Tool Wear Estimation*, IEEE, https://doi.org/10.1109/intercon52678.2021.9532759.