RUL Prediction for Cold Forming Production Tooling

Wim De Mulder and Alessandro Di Bucchianico

Department of Mathematics and Computer Science
Eindhoven University of Technology
Eindhoven, The Netherlands
Email: wim.demulder@ugent.be
a.d.bucchianico@tue.nl

Haije Zijlstra

Philips
Drachten, The Netherlands
Email: haije.zijlstra@philips.com

***Abstract*—This paper presents a work-in-progress contribution that involves a collaboration with Philips, where the goal is to predict the remaining useful lifetime for cold forming production tooling. As the data set is complex, with many outliers and missing data points, we plan to integrate multiple techniques to reliably predict the time until maintenance is required, at least including machine learning methods, bootstrapping and change point detection techniques. The latter two methods are seldomly employed in the domain of remaining useful lifetime prediction, although they deliver very useful additional information compared to mainstream prediction techniques. Despite the fact that times of failure are currently lacking, we were able to perform a useful preliminary data analysis, which resulted in the extraction of several features to be used later as input variables for RUL prediction, and we obtained an interesting unsupervised clustering of a set of selected production runs.**

Keywords–Remaining useful lifetime; Bootstrapping; Change points; Cold forming production tooling.

I. INTRODUCTION

This paper presents a work-in-progress contribution that is part of the ambitious Prophesy project on predictive maintenance [1]. Predictive Maintenance (PdM) refers to the data-driven process of predicting when operational equipment may fail and deploying preventive maintenance to avoid any downtime [2]. The idea is that maintenance should be performed as far in the future as possible, in order to minimize costs, ideally just before serious failure occurs. The predicted output variable by a PdM algorithm is the Remaining Useful Lifetime (RUL) [3], which indicates how long the machine is expected to continue to run without failure. This variable is expressed in units that depend on the application at hand, e.g., the number of cycles that a turbofan engine is expected to run before it breaks down [4].

The goal of the project entails a collaboration with two major industrial companies, namely Jaguar Land Rover and Philips, whose maintenance costs consume a considerable amount of all expenses. In this paper, we describe the Philips use case, involving a very complex data set containing many variables, measured continuously over time, with a lot of missing data points and outliers. Currently, we are focusing on measured force-signals at different angles during the forming process of the involved metal part [5]. The main purpose is to predict the remaining number of die hits (which are accompanied by the generation of a force-signal) that the considered tool is able to produce before breakdown.

Unfortunately, for the considered tool no times of failure are currently available. Nevertheless, the unlabelled large amount of data that has already been collected is suitable for an unsupervised cluster analysis. Furthermore, collaboration with engineers from Philips allowed a useful feature extraction, where the features can later be used as input variables in a supervised RUL prediction method.

The paper is organized as follows. Section II gives some background on the use case, while Section III describes how the (currently unlabelled) data set has been collected. Section IV explains how feature extraction has been performed, despite the fact that times of failure are currently unavailable. Unsupervised cluster analysis of 13 selected production runs has been applied for each of the features, as outlined in Section V. Finally, Sections VI and VII describe two interesting future research lines for this particular use case, but also for RUL prediction in general.

II. DESCRIPTION OF THE USE CASE

Philips Consumer Lifestyle in Drachten develops a wide range of innovative products like rotary shavers, beard trimmers, hairdryers, epilators, vacuum cleaners, SENSEO coffeemakers and Wake-up Lights. Philips Drachten employs 2000 people, amongst which 600 developers with 35 different nationalities. Philips Drachten is also world leader in mass production of rotary shaving devices, occupying over 50% market-share of a €1 billion market.

Multiple production lines take care of cold forming metal parts. Each production line creates a product mix. On these production lines over 50 individual metal products are created. In total, there are over 300 individual dies. The project goal is dedicated to a single production line and a single die set. When this goal will be accomplished, the developed model will be expanded to cover more production lines and more dies.

Tooling maintenance is performed in the tool workshop. A production run for the tool maintenance is triggered by production for the following reasons:

- Production run finished.
- Pre-defined lifetime threshold reached.
- Product quality issue or tool malfunction.

About 600 production runs are triggered yearly. The accuracy and the diversity of the wear parts, together with the interactions between these parts during processing, is a big challenge for maintenance. Therefore, the quality of work is strongly dependent on the skills and craftsmanship of the maintenance engineers. In many cases, highly skilled,

 ISBN: 978-1-61208-831-0

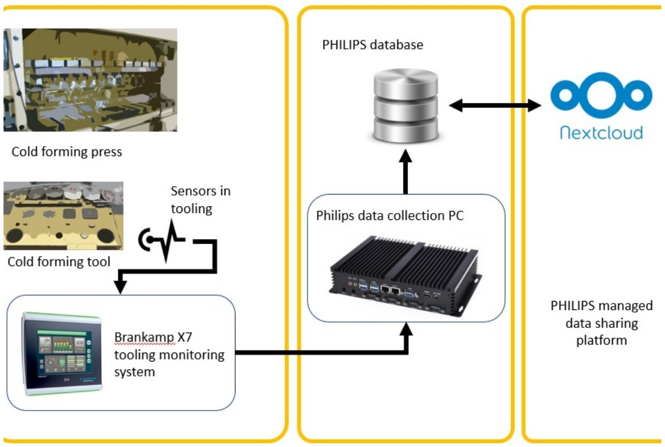

Figure 1. Global overview of the use case.

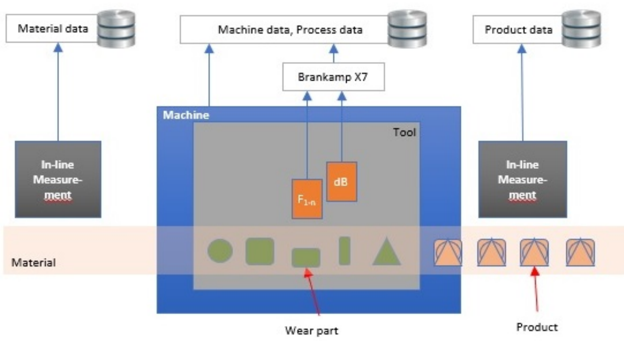

Figure 2. Some technical details of the use case.

second-line support is needed in case of non-standard problem solving. Breakdowns come with high maintenance costs, which explains the need to investigate modern technologies that enable the shift from breakdown maintenance to predictive maintenance.

The Philips predictive maintenance use case is defined around the cold forming tooling for high precision metal parts. Although the cold forming tool consists of a high number of parts, for the sake of clarity in this use case, two wear parts are considered: one cutting punch and one die-plate. Figure 1 gives a global overview of the use case, while Figure 2 displays some more technical details.

III. Data collection

The data set includes the force-signal as registered during the cold-forming operation, and is collected from several sources. To enable data extraction from existing machines, modifications to the machine control systems have been made and additional measurements have been programmed in the in-line measuring machines and a Brankamp X7 process monitoring system [6].

Figure 3 shows such a Brankamp X7 system, as well as some possible process curves. The X7 system allows up to 24 channels for an extended process monitoring. The HMI part of the system runs on a Windows Operation System, so that an easy connection to other Prophesy parts is possible. Furthermore, the X7 Cockpit provides a switchable mask design with flexible arrangement of the monitoring channels (according to the machine configuration). Binary input signals can be monitored with up to three monitoring windows to ensure the

Figure 3. Brankamp X7 process monitoring system.

earliest possible fault detection. The failure distribution shows machine downtimes and the frequency of process failures for a quick and easy failure analysis.

During one stroke of the cold forming press, the cutting force, as measured by a sensor manufactured by United Electric Controls, is stored at 500 measurement points. These 500 measurement points correspond to force-signals at angles ranging from 50 degrees to 110 degrees in steps of 0.12 degrees. At a normal production rate, 1 stroke is stored every 60 seconds, in CSV format, and then pushed to the Philips Manufacturing Execution System, from where it can be downloaded by the Prophesy partners in order to perform data analysis.

IV. Feature extraction

A careful analysis that was performed together with engineers from Philips showed that only 13 production runs that were obtained during the Prophesy project, are eligible to be used as training data. As there are 500 input variables (cf. Section III), it is necessary to perform a dimensionality reduction [7]. Five features were extracted:

1) Variable 1: the 90% percentile force at angle 71.72 degrees. The 90% percentile is taken over windows consisting of 251 time points, as follows: for a given time point, consider the set of forces at 71.72 degrees together with those at the 125 time points to the left and together with those at the 125 time points to the right. From this set the 90% percentile is taken. The reason to consider the 90% percentile is that the maximum force is more interesting than, e.g., the average force, from a mechanical point of view, but because of the presence of outliers the 90% percentile force was taken as a more robust alternative to the maximum force. For time points at the beginning or at the end of a production run, for which less than 125 time points are available to either the left or the right, additional time points from the other side are taken in computing the 90% percentile force.
2) Variable 2: the Area Under the Curve (AUC) [8] of the forces between 70 and 75 degrees.
3) Variable 3: the 90% percentile force at 79 degrees, where the 90% percentile is computed as for variable 1.
4) Variable 4: the AUC of the forces between 78 and 80 degrees.

The features were extracted by analyzing the time series of the forces for the considered production runs, and discussing with engineers from Philips the mechanical meaning and relevance of the observed patterns.

V. Cluster analysis of the production runs

An important disadvantage of the current data set is that no production runs end in failure, implying that RULs are missing. Consequently, supervised machine learning cannot be applied, and the best we can currently do is performing an unsupervised analysis [9]. Cluster analysis [10], and in particular k-means [11], is a typical unsupervised tool to gain some insight into the structure of an unlabelled data set.

The k-means algorithm is applied to the 13 time series that correspond to the production runs, using the Euclidean distance measure. Clustering is repeated for each of the four feature variables. However, because the production runs do not all have the same number of time points, we truncate all production runs at 1640 time points, which is the number of time points of the shortest production run. Unfortunately, the optimal number of clusters is unknown. Therefore, we applied three validation measures that try to detect the optimal number of clusters, using the clValid package from the statistical software R: connectivity, the Dunn index and the Silhouette index [12]. The result is shown in Table I.

Table I. Optimal number of clusters according to three selected cluster validation measures.

	Connectivity	Dunn	Silhouette
Variable 1	2	9	2
Variable 2	2	10	2
Variable 3	2	2	2
Variable 4	2	2	2

Although the measures do not agree on the optimal number of clusters for all variables, it is clear that two clusters is overall the best choice. Table II shows the cluster index for each production run for each of the variables, where clustering is performed with two clusters.

Table II. Clustering of the production runs for each of the feature variables.

	Variable 1	Variable 2	Variable 3	Variable 4
Production run 1	1	1	1	1
Production run 2	1	1	1	1
Production run 3	1	1	1	1
Production run 4	1	1	1	1
Production run 5	1	1	2	2
Production run 6	1	1	2	2
Production run 7	1	1	2	2
Production run 8	1	1	2	2
Production run 9	2	2	2	2
Production run 10	2	1	2	2
Production run 11	2	1	2	2
Production run 12	2	1	2	2
Production run 13	2	1	2	2

It is seen that the clustering is not the same for each of the variables. However, for variables 3 and 4 the clustering *is* exactly the same; these variables group the first four production runs together, while the other production runs belong to the other cluster. Variable 1, on the other hand, considers the first eight production runs as similar, while the other production runs are clustered in another group. Finally, variable 2 seems not to result in a meaningful clustering: all production runs are clustered together, except production run 9.

We can integrate the different clusterings into one summary clustering as follows. Each clustering can be represented as a

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]	[,11]	[,12]	[,13]
[1,]	0.0	0.0	0.0	0.0	0.4	0.6	0.6	0.6	1.0	1.0	1.0	0.6	1.0
[2,]	0.0	0.0	0.0	0.0	0.4	0.6	0.6	0.6	1.0	1.0	1.0	0.6	1.0
[3,]	0.0	0.0	0.0	0.0	0.6	0.4	0.4	0.4	1.0	0.6	0.6	1.0	0.6
[4,]	0.0	0.0	0.0	0.0	0.4	0.6	0.6	0.6	1.0	1.0	1.0	0.6	1.0
[5,]	0.4	0.4	0.6	0.4	0.0	0.0	0.0	0.0	0.6	0.4	0.4	0.0	0.4
[6,]	0.6	0.6	0.4	0.6	0.0	0.0	0.0	0.0	0.4	0.0	0.0	0.4	0.0
[7,]	0.6	0.6	0.4	0.6	0.0	0.0	0.0	0.0	0.4	0.0	0.0	0.4	0.0
[8,]	0.6	0.6	0.4	0.6	0.0	0.0	0.0	0.0	0.4	0.0	0.0	0.4	0.0
[9,]	1.0	1.0	1.0	1.0	0.6	0.4	0.4	0.4	0.0	0.0	0.0	0.4	0.0
[10,]	1.0	1.0	0.6	1.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
[11,]	1.0	1.0	0.6	1.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
[12,]	0.6	0.6	1.0	0.6	0.0	0.4	0.4	0.4	0.4	0.0	0.0	0.0	0.0
[13,]	1.0	1.0	0.6	1.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 4. Average clustering matrix, with values 0.2 rounded to 0, and values 0.8 rounded to 1.

matrix M with elements $M[i, j]$ defined as follows: $M[i, j] = 0$ if production run i and j belong to the same cluster, and $M[i, j] = 1$ otherwise. The four resulting matrices can then be averaged over the number of variables, i.e., each $M[i, j]$ is divided by four. The matrix elements then take values in $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$. For ease of interpretation we round the value 0.2 to 0 and the value 0.8 to 1. The matrix that is obtained in this way, as produced by R, is shown in Figure 4. This matrix provides the following information:

- Production runs 1 to 4 are similar.
- Production runs 5 to 8 are similar.
- Production runs 9, 10, 11 and 13 are similar.
- Production runs 10 to 13 are similar.

The last two observations could also be rephrased as saying that production runs 9 to 13 are similar, with the note that there is some dissimilarity between production runs 9 and 12.

Currently we started discussing if the results of this cluster analysis may be used in future RUL prediction. This requires to analyze whether production runs 1 to 4, production runs 5 to 8, and production runs 9 to 13 share certain properties that might be related to the 'health' of the involved mechanical parts. If so, this knowledge can be used in a later stage, together with times of failure when they become available, to predict the RUL.

VI. Future work: bootstrapping and prediction interval

Applications on RUL prediction often restrict to returning a single RUL value. However, it is very useful to have an indication of the reliability of this prediction, for example in terms of a prediction interval. Although many machine learning techniques, e.g., artificial neural networks, only return a predicted value, it is easy to obtain a prediction interval by means of bootstrapping [13]. By randomly sampling from the training set and training a model on each sample, one obtains a set of models, each having a slightly different view on the data set. Consequently, averaging the predictions of all these models results in a robust prediction, and the set of predictions itself can be considered as a range of possible RUL values that vary with certain peculiarities in the data set. From the set of predictions, which represent a histogram of values, it is very simple to obtain a 95% prediction interval. Noteworthy advantages of bootstrapping are its simplicity in terms of implementation, and the observation that it is asymptotically more accurate than the standard intervals obtained using sample variance and assumptions of normality [14]. Prediction intervals offer more decision-making power to industrial partners. For example, very conservative engineers will be inclined not to rely on

 ISBN: 978-1-61208-831-0

the predicted value, but on the left endpoint of the prediction interval when scheduling maintenance.

Even if prediction intervals are taken into account, it is often neglected to evaluate them. That is, evaluation of a prediction model is often restricted to the predicted values. However, validation measures have been developed to evaluate prediction intervals. For example, the interval score [15] rewards narrow intervals, while at the same time penalizing lack of coverage. It is defined as

$$\Big(u(\mathbf{x}) - l(\mathbf{x})\Big) + \frac{2}{\alpha}\Big(l(\mathbf{x}) - \nu(\mathbf{x})\Big)\mathbf{1}_{\{\nu(\mathbf{x})<l(\mathbf{x})\}} + \frac{2}{\alpha}\Big(\nu(\mathbf{x}) - u(\mathbf{x})\Big)\mathbf{1}_{\{\nu(\mathbf{x})>u(\mathbf{x})\}}$$

for a $(1-\alpha)\%$ prediction interval $[l(\mathbf{x}), u(\mathbf{x})]$, where $\mathbf{1}_{\{expr\}}$ refers to the indicator function, being 1 if expression $expr$ holds and 0 otherwise. In our view, implementations related to RUL prediction should always produce a prediction interval, and these prediction intervals should be taken into account when performance is evaluated by comparing to the results of other models. We will do so in our future research on the Philips use case.

VII. Future work: change point detection

Change point detection refers to identifying when certain properties of a probability distribution, in particular the mean and the variance, of a time series change [16]. As far as we know, change point detection methods are rarely used in RUL prediction. This is unjustified, as such methods are complementary to traditional RUL prediction methods: whereas most RUL methods, e.g., artificial neural networks, implicitly assume continuity in the time series, change point detection is able to handle sudden discontinuities or trends. In this way, it might prove useful to combine traditional machine learning techniques with change point detection methods in RUL prediction. For example, as long as no change point is detected, engineers might rely on the midpoint value of the prediction interval for maintenance. However, whenever a change point is notified, indicating a change in regime (possibly due to the failure of a non critical part of the involved tool), engineers might react properly by relying on the left endpoint of the prediction interval instead, thereby discounting the danger of a premature failure.

We experimented with the package 'changepoint' in R [17]. For example, Figure 5 shows the forces during a selected production run at angle 71 degrees, together with the jumps in an otherwise constant mean trend. One changepoint (jump) was detected, as seen from the figure.

Change points deliver additional information compared to mainstream RUL predictors. For our case study, we envisage developing a measure that takes into account the number of angles at which a change point arises: if a change point occurs at a high number of angles at the same time, this indicates a more serious warning compared to a change point occurring at only one or a small number of angles at the same time. In the first case, a more conservative RUL should be predicted than in the second case.

VIII. Conclusion

In this work-in-progress contribution we have described a use case from a cold forming production tooling process that involves collaboration with Philips in the context of a major EU project named Prophesy. The purpose is to predict the remaining useful lifetime of certain tools that come with high maintenance costs. Our plan is to combine several techniques in order to fulfill this goal:

- Machine learning techniques, in particular artificial neural networks, to produce the RUL.
- Bootstrapping, to obtain a prediction interval for the predicted RUL.
- Change point detection, to ensure that not only gradual changes in performance of the tool are detected, but that also more abrupt changes are identified.

An important challenge will be to integrate all these methods into a reliable and efficient RUL prediction mechanism.

Bootstrapping and change point detection are often neglected in RUL prediction, although these techniques create additional relevant information on the predicted RUL. Perhaps these methods just need to find their way in the domain of RUL prediction. Our work might be an incentive for other researchers in RUL prediction to consider the use of these techniques.

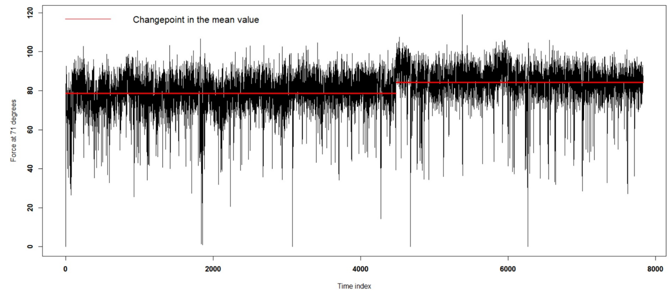

Figure 5. Change point detection for the mean during a certain production run for forces measured at angle 71 degrees.

Acknowledgment

This work was supported by European Union's Horizon 2020 grant no.766994 (PROPHESY).

References

[1] Prophesy, https://prophesy.eu/, last access: August 20, 2020.

[2] J. Levitt, Ed., Complete Guide to Preventive and Predictive Maintenance. Industrial Press, 2011.

[3] X. Si, W. Wang, C. Hu, and D. Zhou, "Remaining Useful Life Estimation – A Review on the Statistical Data Driven Approaches," European Journal of Operational Research, vol. 213, 2011, pp. 1–14.

[4] V. Mathew, T. Toby, V. Singh, M. Rao, and M. Kumar, "Prediction of Remaining Useful Lifetime (RUL) of Turbofan Engine Using Machine Learning," in Proceedings of the IEEE International Conference on Circuits and Systems (ICCS), 2007, pp. 306–311.

[5] I. Suchy, Ed., Handbook of Die Design. McGraw-Hill Education, Jan. 2005.

[6] Brankamp, http://www.brankamp.com/en/index.php?page=brankamp_x7_artificial_intelligence_is_the_key, last access: August 20, 2020.

[7] H. Lu, K. Plataniotis, and A. Venetsanopoulos, Eds., Multilinear Subspace Learning: Dimensionality Reduction of Multidimensional Data. Chapman and Hall/CRC, Jan. 2013.

[8] A. Bradley, "The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms," Pattern Recognition Letters, vol. 30, 1997, pp. 1145–1159.

[9] M. Kyan, P. Muneesawang, K. Jarrah, and L. Guan, Eds., Unsupervised Learning: A Dynamic Approach. Wiley-IEEE Press, Jan. 2014.

[10] D. Xu and Y. Tian, "A Comprehensive Survey of Clustering Algorithms," Annals of Data Science, vol. 2, 2015, pp. 165–193.

[11] A. Jain, “Data Clustering: 50 Years Beyond K-means,” Pattern Recognition Letters, vol. 31, 2010, pp. 651–666.

[12] G. Brock, V. Pihur, S. Datta, and S. Datta, “clValid: An R Package for Cluster Validation,” Journal of Statistical Software, vol. 25, 2008, p. i04.

[13] J. Shao and D. Tu, Eds., The Jackknife and Bootstrap. Springer, Jan. 1996.

[14] T. DiCiccio and B. Efron, “Bootstrap Confidence Intervals,” Statistical Science, vol. 11, 1996, pp. 189–228.

[15] T. Gneiting and A. Raftery, “Strictly proper scoring rules, prediction, and estimation,” Journal of the American Statistical Association, vol. 102, 2007, pp. 359–378.

[16] S. Aminikhanghahi and D. Cook, “A Survey of Methods for Time Series Change Point Detection,” Knowledge and Information Systems, vol. 51, 2017, pp. 339–367.

[17] R. Killick and I. Eckley, “changepoint: An R Package for Changepoint Analysis,” Journal of Statistical Software, vol. 58, 2014, pp. 1–19.

 ISBN: 978-1-61208-831-0