Predictive Maintenance Approaches with Free-text Labels: A Case Study in the Oil Industry

Duarte Oliper, [1] **Vitor Rolla,** [1] **Tomás Souper** [1]

[1] Fraunhofer Portugal AICOS
duarte.pereira@fraunhofer.pt, vitor.rolla@fraunhofer.pt, tomas.pereira@fraunhofer.pt

Abstract

Predictive maintenance is desired by industries that require optimization of processes to prevent unnecessary costs in maintenance. Monitoring sensors are widely applied, but the lack of structured data and annotations can limit domain understanding when training intelligent algorithms for failure prediction. This study focuses on a real-world oil and gas industry dataset, with multivariate time series data and manual text annotations of maintenance periods. After iterative experimentation, the better approach utilizes the Random Forest classifier, with eleven failures clustered through an unsupervised brute-force technique, achieving 73% accuracy, 76% precision, and 70% recall. Tackling real-world predictive maintenance through multiple approaches is the path to a successful solution.

Motivation

Predictive Maintenance (PdM) consists of diagnosing failure signs within monitoring data, allowing for early detection and resolution of the issue. Previously there were two common ways; in preventive maintenance (PvM), the maintenance is scheduled with a fixed period, while in reactive maintenance (RM), the maintenance is done after the failure happens. The optimal detection of PdM reduces the material, and downtime costs, leading to process optimization (Pech, Vrchota, and Bednář 2021). According to Deloitte insights (Deloitte 2017), PdM allows an increase in equipment uptime by up to 20%, a maintenance cost decrease of up to 10%, and to deploy industry resources more efficiently and effectively.

Figure 1 describes an overall relation between costs and the condition. The lines represent the full asset as one (e.g., the whole production Machine), while the dashes represent a single component. In PvM, the maintenance cost is high, while the repair cost is low. In contrast, in RM, the repair cost is higher. PdM tries to balance both of these approaches. Maintenance can be related to sustainability, e.g., amount of devices, waste, energy spent, etc. When isolating, only the defective looks better to do RM (dashed lines): less used components are used as much as possible. However, when analyzing the overall production, PvM has a better energy use, while RM has a better use of the component lifetime. PdM can find a balance between the advantages of both strategies, e.g., savings in energy (PvM) and less waste (RM).

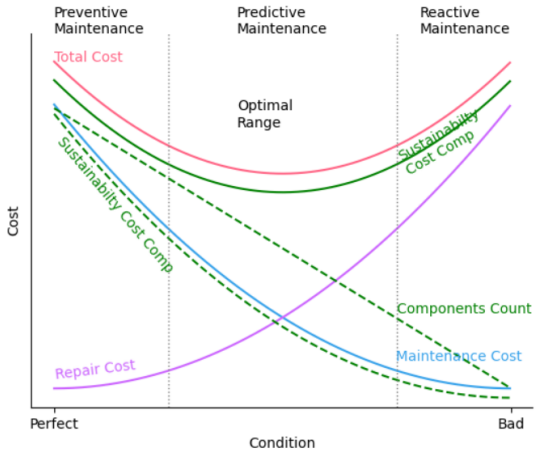

Figure 1: Relation Between Cost and Asset condition

One of the main challenges of applying PdM in a real industry process is the lack of structured data and annotations, which limits domain understanding.

Materials and Methods

This study follows a collaboration proposal from an undisclosed oil and gas company. The main goal of the collaboration is to recognize failure signs within a distillery asset and provide valuable historical and real-time machine learning for a future process dashboard.

Dataset

A real-world dataset of the oil industry is explored in this study, including multivariate time series data of monitoring sensors with hourly resolution, and manual annotations of maintenance periods with daily resolution.

Regarding the annotations, three types of stops were identified: (i) scheduled, (ii) external, and (iii) failure. Scheduled maintenance does not indicate that the equipment needs care; it is purely recurrent. However, such a type of maintenance should bring the equipment to an optimal working state in most cases; therefore, it is not beneficial to failure prediction directly. Nevertheless, they store an understanding of the behaviors of a healthy device. Examples of external faults are power or supply interruptions, which are not helpful for predictive maintenance. Finally, twenty failures

were eligible for this research. The main criterion for the selection of failures was healthy operation time prior.

Data Preprocessing

Preprocessing of the multivariate data starts with null value interpolation. Based on domain experts' annotations and intrinsic knowledge, it was determined that elected failures should have at least ten-day of plain functioning before failure. After the Remaining Useful Life (RUL) computation, i.e., the number of days until each failure, only twenty failures had the ten-day rule prior; those were incorporated in this study. The binary label was defined with a five-day threshold according to domain experts' experiences. Ergo, the data was labeled as follows:

- 1 for values of RUL lower than five days before failure.
- 0 for values of RUL larger than five days.

The threshold was set to half the window value to prevent class imbalance and ease metric analysis. Data was normalized through standard scaling fit to the train data.

Feature extraction with the Time Series Feature Extraction Library (TSFEL) (Barandas et al. 2020) added over 60 statistical, temporal, and spectral features from time series data per window.

Experiments

After determining the twenty stops that made sense to this study, the following experiments were carried out. First, supervised learning with a tree-based algorithm - Random Forests (RF) - and a recurrent neural network - Long Short Term Memory (LSTM). Second, unsupervised learning based on text similarity over the annotated dataset, followed by anomaly detection (isolation forest and local outlier factor algorithms) combined with TSFEL over the time series. Finally, a brute force approach was used to determine the most similar failures.

The asset's RUL is framed as a classification problem and estimated using Tree-Based and LSTM models. After initial experimentation, suspicions arose about the existence of different failure types. A fact that led to this insight was the improved performance when using a 50% train-test split compared to 25%; see Table 1.

The clustering of different types of stops was studied from different unsupervised perspectives. For instance, the datasets' unstructured annotations of failures led to a Natural Language Processing (NLP) text similarity approach (Figure 3 at the Supplementary Materials). Furthermore, anomaly detection algorithms, such as isolation forest and local outlier factors were employed to remove the less similar stops, to make the large group more concise.

Finally, a random brute force approach resulted in a concise group of 11 similar failures. Clustering based on brute force was performed through the selection of stops that contributed most to a better performance of the models after hundreds of experiments with random datasets' train/test splits.

Due to project needs, a prototype of a web dashboard illustrating the practical difference between reactive and predictive maintenance was developed. (Figure 2 at the Supplementary Material).

Results

In Table 1, LSTM results show that the proportion chosen for the test set influences model performance. For a usual 25% test split, all metrics have lower values than for the 50% test split. This may indicate an imbalanced set of failure types. Regarding the RF results, metrics increased after performing unsupervised failure clustering.

	Test Size	No. Failures	Accuracy	Precision	Recall
LSTM	25	20	46	47	42
	50	20	56	56	67
RF	25	20	59	60	61
	25	11	73	76	70

Table 1: LSTM and Random Forest results, in percentages. Reliance in train-test proportions in LSTM performance. The positive impact of failure clustering in Random Forest performance.

When compared to the LSTM, RF attained better results since neural networks rely on large amounts of data to perform well in similar tasks. Therefore it is possible to conclude that the better approach utilizes the Random Forest classifier, after brute-force clustering, with 73% accuracy, 76% precision, and 70% recall.

Conclusion

In this paper, the problem of predictive maintenance is framed from diverse angles, including supervised, unsupervised, and natural language processing. Failure clustering methods allow for the acquisition of more robust labels for further modeling and understanding of the processes. In a perfect scenario, annotations and precise maintenance periods should be automated and standardized at the industry level to ease the adoption of predictive maintenance algorithms.

This study shows the feasibility of applying these approaches to many industries interested in acquiring the benefits of predictive maintenance.

Acknowledgements

This article is the result of work conducted under the project "NewGenTSFAE: New Generation Test Systems for Future Automotive Electronics" (no. 46990), supported by the Competitiveness and Internationalisation Operational Program, Portugal (POCI), and the "I&DT em Copromoção" (R&D and Promotion) program, through the European Regional Development Fund (ERDP).

References

Barandas, M.; Folgado, D.; Fernandes, L.; Santos, S.; Abreu, M.; Bota, P.; Liu, H.; Schultz, T.; and Gamboa, H. 2020. TSFEL: Time Series Feature Extraction Library. *SoftwareX*, 11: 100456.

Deloitte. 2017. Predictive maintenance and the smart factory. Accessed: 2022-11-17.

Pech, M.; Vrchota, J.; and Bednář, J. 2021. Predictive Maintenance and Intelligent Sensors in Smart Factory: Review. *Sensors*, 21(4): 1470. Number: 4 Publisher: Multidisciplinary Digital Publishing Institute.

Supplementary Material

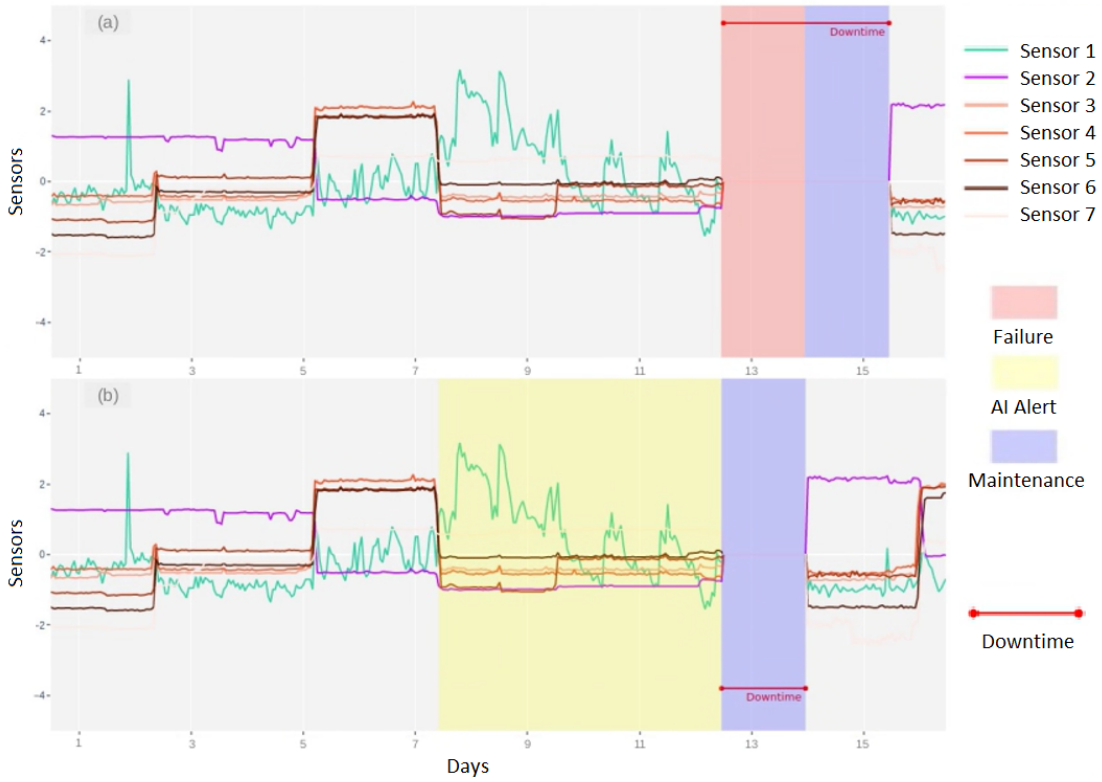

Figure 2: Illustration of downtime difference in maintenance procedures. In reactive maintenance (above), maintenance only occurs after critical failure, with increased downtime. In predictive maintenance (below), the AI alerts for early maintenance scheduling, preventing critical failure and associated costs with additional repairs and downtime.

.

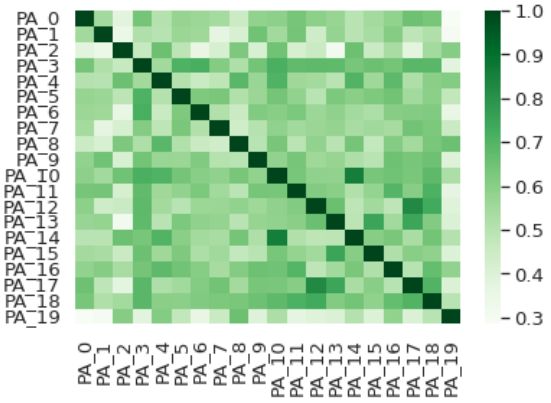

Figure 3: Text similarity results. NLP showed a clear similarity between stops 12-17 and 10-14. Stops 2 and 19 showed high individuality, corroborating the hypothesis that a small test set could be biased.

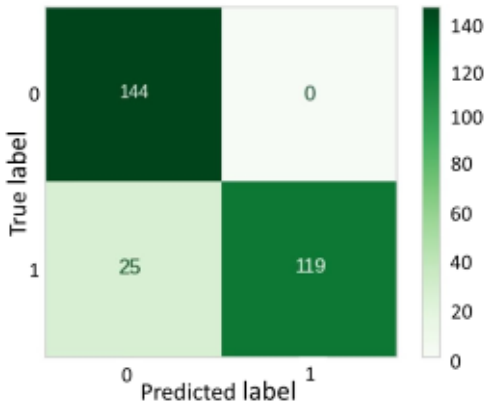

Figure 4: This confusion matrix refers to the AI model presented in Figure 2. In this case, the results are optimistic (accuracy: 91%, precision:100%, recall: 83%) because they refer to the best fold in cross-validation; when all stops were used in training except the one in the test.