Predictive maintenance of a rotating condenser inside a synchrocyclotron

Valentin Hamaide and François Glineur

ICTEAM, UCLouvain, Belgium
valentin.hamaide@uclouvain.be
francois.glineur@uclouvain.be

Abstract. This paper investigates data-driven methods to predict failures of a rotating condenser (RotCo) inside a synchrocyclotron for a proton therapy treatment system [12]. Downtime caused by a failure of the system can lead to significant delays in the treatment of the patients, which is why having a reliable predictive maintenance system is essential. The condenser rotates at high speed and rolling bearing elements are responsible for maintaining low friction between the moving components. The aim is to predict failures of the bearing box which contains the shaft and the bearing elements. Several sensors within the cyclotron are constantly measuring multiple relevant signals but, notably, vibration data is not available. We leverage those time-series data to predict a few days in advance whether a failure is likely to happen. To do this, we propose a two-level approach that relies on combining the output of a classifier with an aggregator based on a custom business metric specifically designed for this problem.

Keywords: Predictive maintenance · Rotating machine · Machine learning · Time series

1 Introduction to our predictive maintenance problem

Downtime caused by the failure of a component of a proton therapy system can lead to delays in the treatment of patients. We investigate one of the component subjected to failure, called the RotCo (for Rotating Condenser) and try to predict a few days in advance whether a failure is likely to happen.

The RotCo is composed of a stator and a rotor with eight blades and rotates at a constant speed of 7500 RPM, giving a repetition rate of 1kHz. We are interested in predicting the state of a specific sub-device of the RotCo, the bearing box which contains the bearing elements, that shows signs of weaknesses visible to operators only few hours before the failure happens. A vast scientific literature exists for the diagnostics and prognosis of bearing elements. However, nearly all the research of the subject is based on vibration data, which are not available in our situtation. Hence, we must follow a different kind of approach: we chose a purely data-driven framework based on machine-learning.

Eight sensors constantly measure the health of the RotCo at a rate of $1Hz$. The signals measured and their descriptions are shown in Table 1 below.

Table 1: Description of available signals.

Name	**Description**
Speed	Speed at which the rotor is rotating
Torque	Torque applied to the rotor
Current	DC component of the current
Voltage	DC component of the voltage
Bearing Temperature	Temperature inside the bearing box
Motor Temperature	Temperature of the motor
Pyrometer Temperature	Temperature of the rotor measured via infrared.
Vacuum	Pressure inside the vacuum chamber (vacuum measurement is impacted when the bearing box friction generates gas release).

From those time-series signals, we want to predict whether the machine is going to fail in the next days ahead. The business need is to detect a failure five days in advance (seven days being even better) so that if a sign of failure is detected on a Monday, a maintenance can be scheduled in the week-end to limit the downtime. Based on theses business considerations, a specific metric was designed to evaluate the performance of a given prediction method and is depicted in Fig. 1. It provides a score between zero and one (higher is better) computed from a piecewise-linear function of the number of days between the prediction and the actual failure:

– A prediction 7 days ahead gives a perfect score.
– Predictions between 7 and 0 days assign a score that decreases linearly to zero (with a slightly higher slope between 7 and 5 days).
– Similarly, predictions ranging from 7 to 15 days are assigned a score linearly decreasing from 1 to 0 (with a slightly higher slope between 7 and 10 days).
– Finally, predicting a failure more than 15 days in advance leads to a zero score to reflect the unexploited lifetime.

2 Previous work

There exists multiple ways to tackle a predictive maintenance problem, coming from different research fields and answering different kind of questions. A large number of works can be found in the literature, and many surveys are available: Jardine et al. [6], Peng et al. [13], Ahmad and Kamaruddin [1] (the latter focusing on signal monitoring methods), or more recently [14] from Shin and Jun. Those reviews split the predictive maintenance program into essentially three parts: data acquisition, data processing and maintenance decision making.

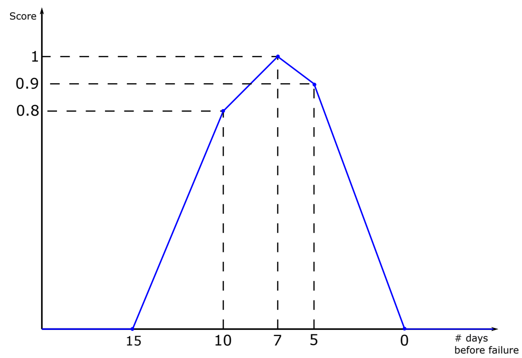

Fig. 1: Business metric used to evaluate the performance of failure predictions.

Viewed at a conceptual level, techniques for maintenance decision making can be split into three categories:

1. Anomaly detection techniques [2] consist in attempting to flag anomalous behavior and isolate or identify faults, and is sometimes referred to as diagnostics in the literature.
2. Techniques based on classification [16] try to predict whether a system is going to fail within a given time-window.
3. Techniques based on regression try to estimate the Remaining Useful Life (RUL), i.e. how much lifetime remains before a failure, and is sometimes referred to as prognostic in the literature [5, 9].

RUL estimation usually involves making an assumption of the degradation process via health indicators (directly observed or created via feature engineering) and predict the remaining lifetime based on this assumption. We do not make any assumption on the degradation process and therefore exclude this approach. Then, as our metric for prediction quality clearly involves time, we choose the classification approach over anomaly detection, where we will try to predict whether a system is going to fail within a given time horizon. To fit the classification setting we will label training data points as nominal or critical, depending on whether failure occurs within that time horizon.

A second aspect, orthogonal the the above discussion, is the type of model used. The literature distinguishes between physics-based models (e.g. [8], [11]), knowledge based models (expert & fuzzy systems, e.g. [4]), threshold-based signal processing (e.g. [1]), statistics models (e.g. [15] for a survey) and machine learning (ML)/Artificial intelligence (AI) models (e.g. [10]). We decided to choose the machine learning approach because the system appears to be too complex to be modelled from a physical point of view, and we have enough data to feed into a ML model. We do not consider the threshold-based signal processing approach as it is usually based on the vibration data that we lack.

One interesting point is the fact that none of the above-mentioned surveys – nor specific applications in the literature – seem to describe a classification approach based on a time horizon before the failure. On the one hand, classification is usually used in a detection problem based on classifying different faults or discriminating between a faulty system and an healthy one without notion of time. On the other hand, when time is taken into account it seems to be always modelled as a regression problem implementing the RUL. We found one instance of previous work involving classification with a time-horizon for an electronic device, that can be found in [16]. Their approach consists of training multiple classifiers with different predictive horizons and chose the one that minimizes a cost function. The device is then replaced when the optimal classifier detects a faulty data point. The approach we describe is somehow comparable, with our business metric replacing their cost function, albeit we combine our classifier with a second-level aggregator (see Section 3.3).

Specific predictive maintenance techniques dedicated to rotating machines and bearing elements also exist but almost always rely on vibration data which, as mentioned before, are not available. For an overview on maintenance techniques for rotary machinery, the reader can refer to [3, 7].

3 Description of our approach

3.1 Data acquisition

We collected about a year of data during which 9 replacements occurred. From those 9 replacements, 8 were real failures and 1 was a preventive replacement.

Due to the high number of data points (for one year of data: about 30 millions points per signal), we decrease their number by computing averages over 30-second intervals, leading to about 500,000 data points per signal which is more tractable. All experiments will be run using those.

3.2 Data processing and feature design

After cleaning the data and removing physically impossible values (due to sensor errors), we create some new features in the time and frequency domain. Those features are computed on adjacent non-overlapping 2-hour long time windows, separately for each signal.

The time domain features used for this research are computed as follows: for each signal, the following are computed over each time window (240 samples): mean, standard deviation, skewness, kurtosis and sum of peaks, which is the number of peaks occurring in a time window. A peak is defined as a local maximum (by comparison with its two neighbouring values) that is greater than $\text{median}(X) + 2\text{mad}(X)$, where $\text{median}(X)$ and $\text{mad}(X)$ are respectively the median and the median absolute deviation of the window.

The frequency domain features are computed via Fourier transform of the time window values (which contains 240 samples). Those amplitudes are split into five frequency bins (i.e. 48 samples). Then, in each of those five bins, we compute the average of the five largest amplitudes, and use those as features. This procedure is used as a tradeoff between using the average and the peak amplitude inside of each bin.

Independently of how those features are designed, we investigate two different ways of handling each time-series: a simple approach where each data point contains all features computed over a single two-hour time window for each signal, and a history-enriched approach where we gather for each signal the twelve most recent values of each feature (i.e. going 24 hours backwards in time), with the objective of including enough details about the history without adding too many variables (to avoid risking overfitting and/or excessive computational burden). The simple approach thus contains $8 \times 10 = 80$ features (8 signals and 10 features, being the 5 statistical and 5 frequency features) while the history-enriched case contains $8 \times 10 \times 12 = 960$ features (8 signals, 10 features and 12 time windows).

3.3 Two-level learning approach

Architecture description
We chose a two-level approach, depicted in Fig. 2, that consists of a support vector machine (SVM) binary classifier at the lower level, trained to distinguish between nominal and critical time windows, combined with a higher-level aggregator. A Radial Basis Function (RBF) kernel is chosen for its general-purpose and good accuracy. A one-level approach based on a multi-classifier SVM, where each class corresponds to a certain range of days before failure, was also a possibility but would not have been fully compatible with the considered business metric.

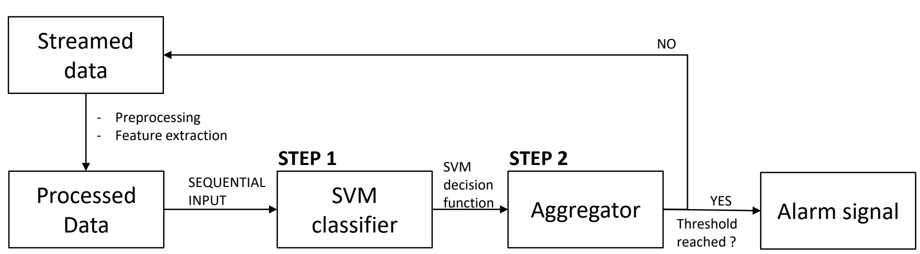

Fig. 2: Two-level approach

Data is first preprocessed and the features are extracted (see section 3.2 above). The processed data is then sent as input to the SVM (step 1 in Fig. 2) that classifies whether the data is critical or nominal.

The result of the SVM is then fed into the aggregator whose responsibility is to decide whether to trigger an alarm or not. The output of the SVM fed into the aggregator is not just its binary class output, but the distance to the hyperplane defined by the decision function

$$f(x) = \sum_{i=1}^{n} \alpha_i K(x_i, x) + b$$

where $K(x_i, x) = \exp(-\gamma \|x_i - x\|^2)$ is the RBF kernel used in the SVM. The aggregator computes a rolling average of the SVM decision function over 8 hours, and raises an alarm if this average exceeds a predefined threshold. Based on the distance interpretation of the decision function (margin) we selected a zero value for the threshold.

Training phase and labelling

For training the SVM, we need to define a labelling. As explained in the introduction, we label each data point as nominal or critical based on whether a failure occurred during some fixed time horizon after the considered time window. As we do not have a precise idea of when the machine actually enters a critical state (which may not even correspond to an actual physical condition), we choose the length of that horizon somewhat arbitrarily to be equal to five days. All data points between the failure date and 5 days before the failure date will be marked as critical and the remaining points as nominal. Other lengths for the labelling horizon have been tested in a preliminary analysis but performed slightly worse.

The general architecture of the training process is depicted in Fig. 3. After data is processed and labelled, it is split in 9 folds, where each fold is a run-to-failure (or a run-to-preventive replacement in one case). Each fold is thus once used as a test set. The SVM is trained on all other folds (STEP 1) and is used to predict on the remaining fold. The output of the classification of this fold is fed to the aggregator that triggers an alarm if it exceeds the zero threshold. As soon as the alarm is triggered, we compute its difference in time with the actual failure, from which we compute the business metric according to Fig.1. The business score is then averaged on the 9 test folds.

Dimensionality reduction

To reduce the dimension of the problem, we use a backward feature selection process with a wrapper approach. We start with all features included in the model and recursively eliminate among all remaining signals or aggregation types the one that increases the score the most (which indicates some level of overfitting for the model with all features). We do this until the score starts to decrease. A cross-validated grid search is done concurrently with the feature selection. The

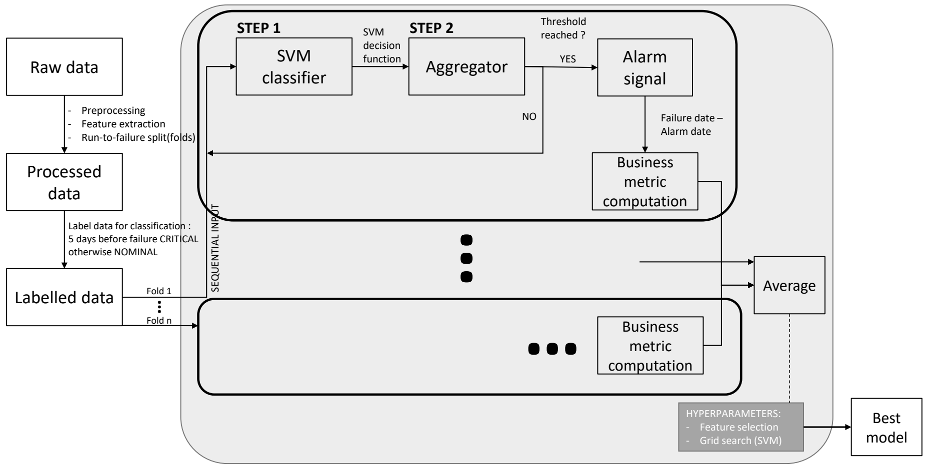

Fig. 3: Training process

principle is the following: We start with all features and recursively eliminate one signal or one aggregation type, i.e. the candidates for removal are Speed, Torque, Current, Voltage, Bearing Temperature, Motor Temperature, Pyrometer temperature, Vacuum, mean, std, skewness, kurtosis, sum of peaks and frequency bins 1,2,3,4,5. This gives a total of 18 candidates. Once we remove a candidate, we remove all the features associated with it. When a signal is eliminated, we remove all its aggregation and history, and when deleting an aggregation type, we remove it in all signals and history. Hence, we remove multiple features per selection step. The candidate removed is selected so as to maximize the business metrics. The backward selection is stopped whenever the removal of a candidate decreases the score by more than one percent.

This process is computationally intensive. Indeed, the first step of the feature selection requires 18 evaluations on different sets of features and, for each set of features, a grid search on the parameters of the SVM with a 9-fold cross-validation is performed. Hyperparameters of the SVM are the penalty of misclassifying data points, C and the parameter of the RBF kernel γ. The grid chosen is a logarithmic grid with the following values: $C = [1, 10, 100, 10^3]$, and $\gamma = [10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}]$. Thus, the first step of the selection alone already requires 18 feature sets $\times$ 16 parameters $\times$ 9 folds $= 2,592$ fits. For each set of features, we thus have 144 scores (from the grid search). For each hyperparameter set, we take the average value across all folds and the score of the feature set is chosen as the maximum value across the hyperparameters.

4 Results

4.1 Feature selection

The results of the feature selection (with a cross-validated grid search) for the history enriched case are shown in figure Fig. 4 (Top). Each point is an average across 9-folds of the best SVM's hyperparameters. As can be seen, removing a candidate (there are several features per candidate) increases the business score up to a certain value around 0.77 and finally starts to decrease after removing 11 candidates. The fact that the score increases is due to overfitting when too many features are taken into account. This also means that not all features are relevant. For example, the voltage is unnecessary and even detrimental to the classifier, as removing this feature already increase the score by 0.13 in comparison with all features taken into account. The remaining features after the selection process are the following: Current, Bearing Temperature and Pyrometer Temperature with the following aggregations: peaks, frequency bins 1 and 5, and standard deviation. This is a count of 144 (3 signals $\times$ 4 aggregations $\times$ 12 windows) features out of the initial 960. The grid search of the hyperparameters of the SVM for the final feature selection is shown in Fig. 4(Bottom).

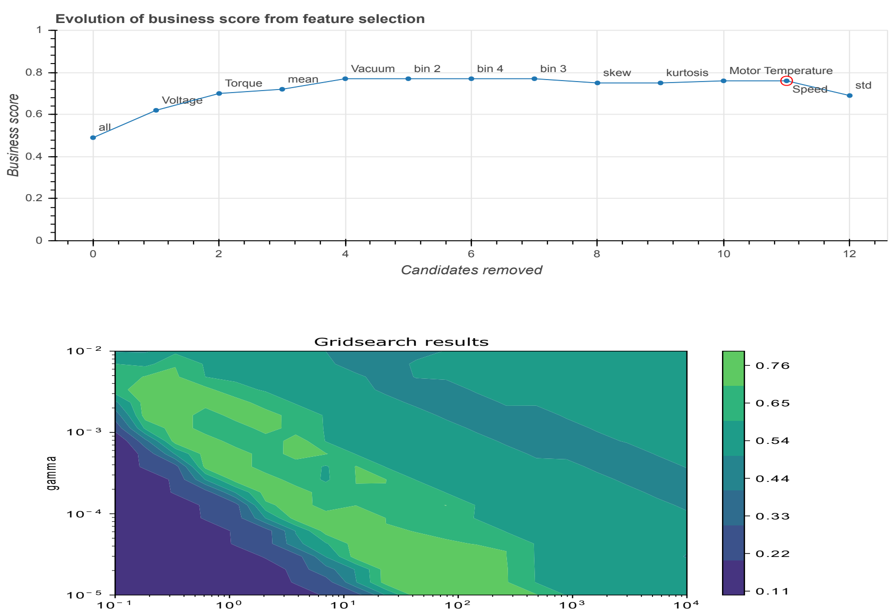

Fig. 4: (Top) Feature selection results (Bottom) Result of the grid search for the final feature selection: business score w.r.t. γ and C SVM hyperparameters.

4.2 Comparison between simple and history-enriched approach

Table 2 shows results for both the simple approach (no history) and the history enriched case, after feature selection and the hyperparameter grid search. Column # days shows how many days in advance an alarm is triggered before the system fails. Column Business shows the business score associated with it (as in Fig. 1). Precision and recall scores (based on the 5-days labelling) are also displayed. As we can see, the average business score for the history enriched approach is 0.2 above the simple approach. Adding history (i.e. data from past time windows) is therefore clearly beneficial.

Table 2: Results

	Without history				With history			
Fold	**# days**	**Business**	**Precision**	**Recall**	**# days**	**Business**	**Precision**	**Recall**
1	2.42	0.43	1	0.46	2.63	0.47	0.96	0.57
2	0.5	0.1	1	0.16	4.17	0.75	1	0.55
3	17.13	0	0.31	0.09	18.96	0	0.32	0.26
4(P)	NA	1	NA	NA	NA	1	0	NA
5	4.08	0.73	1	0.16	4.21	0.76	1	0.78
6	NA	0	0	0	7.92	0.94	0.08	0.02
7	6.54	0.97	0.75	1	6.54	0.98	0.61	0.32
8	7.54	0.96	0	0	8.21	0.91	0.52	0.72
9	5.87	0.94	0.42	0.08	6.92	0.99	0.42	0.33
Mean	**5.51**	**0.57**	**0.56**	**0.24**	**7.44**	**0.76**	**0.61**	**0.44**

The cross-validated result on each fold for the history enriched case is shown in Fig. 5. For each fold, we plot the decision function from the SVM which gives the distance to the separating hyperplane. Whenever a data point is below zero, it is considered a nominal value and whenever it is strictly above zero, it is considered a critical point. The output of the aggregator is shown as the red solid line on the graph. Whenever this aggregate exceeds 0, we send an alarm. As can be seen in the graph, an alarm is sent for all the run-to-failure occurrences except the fourth graph which represents a preventive maintenance. Looking at the results, we observe that our approach performs quite well. Indeed, in 8 out of 9 cases, the algorithm detects a failure between 2.5 and 8 days in advance (while in the remaining case it is detected 19 days in advance, which is slightly too early for our business metric). The algorithm is therefore capable of distinguishing abnormal behaviour and sending a preventive alarm several days ahead of failures.

5 Conclusion and future works

We have shown that it is possible to mostly predict the failure of the rotating condenser in a synchrocyclotron several days in advance, using a purely data-driven approach based on the sampled signals that were provided to us. Indeed, we observed a relatively low false alarm rate, as no false alarm was raised in 8 cases out of 9, and the only case for which the alarm was triggered too early was at roughly 70% of the machine's life, which remains acceptable. We achieved this goal thanks to an original two-level approach based on the combination of a conventional binary classifier (SVM) and an aggregator, optimized with respect to a custom business metric.

Comparing the performance of our approach with a one-step approach based on a multi-classifier SVM is left for further research. The use of transfer learning techniques to apply the learned model to other but similar machines could also be investigated.

References

1. Ahmad, R., Kamaruddin, S.: An overview of time-based and condition-based maintenance in industrial application. Computers & Industrial Engineering **63**(1), 135–149 (2012)
2. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM computing surveys (CSUR) **41**(3), 15 (2009)
3. El-Thalji, I., Jantunen, E.: A summary of fault modelling and predictive health monitoring of rolling element bearings. Mechanical systems and signal processing **60**, 252–272 (2015)
4. Garga, A.K., McClintic, K.T., Campbell, R.L., Yang, C.C., Lebold, M.S., Hay, T.A., Byington, C.S.: Hybrid reasoning for prognostic learning in cbm systems. In: 2001 IEEE Aerospace Conference Proceedings (Cat. No. 01TH8542). vol. 6, pp. 2957–2969. IEEE (2001)
5. Gebraeel, N., Lawley, M., Liu, R., Parmeshwaran, V.: Residual life predictions from vibration-based degradation signals: a neural network approach. IEEE Transactions on industrial electronics **51**(3), 694–700 (2004)
6. Jardine, A.K., Lin, D., Banjevic, D.: A review on machinery diagnostics and prognostics implementing condition-based maintenance. Mechanical systems and signal processing **20**(7), 1483–1510 (2006)
7. Lee, J., Wu, F., Zhao, W., Ghaffari, M., Liao, L., Siegel, D.: Prognostics and health management design for rotary machinery systems—reviews, methodology and applications. Mechanical systems and signal processing **42**(1-2), 314–334 (2014)
8. Li, Y., Billington, S., Zhang, C., Kurfess, T., Danyluk, S., Liang, S.: Adaptive prognostics for rolling element bearing condition. Mechanical systems and signal processing **13**(1), 103–113 (1999)
9. Liao, L., Köttig, F.: Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction. IEEE Transactions on Reliability **63**(1), 191–207 (2014)
10. Liu, R., Yang, B., Zio, E., Chen, X.: Artificial intelligence for fault diagnosis of rotating machinery: A review. Mechanical Systems and Signal Processing **108**, 33–47 (2018)
11. Luo, J., Bixby, A., Pattipati, K., Qiao, L., Kawamoto, M., Chigusa, S.: An interacting multiple model approach to model-based prognostics. In: SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme-System Security and Assurance (Cat. No. 03CH37483). vol. 1, pp. 189–194. IEEE (2003)
12. Pearson, E., Abs, M., Henrotin, S., Kleeven, W., Van de Walle, J., Verbruggen, P., Zaremba, S.: The new IBA superconducting synchrocyclotron (S2C2): From modelling to reality (08 2013)

13. Peng, Y., Dong, M., Zuo, M.J.: Current status of machine prognostics in condition-based maintenance: a review. The International Journal of Advanced Manufacturing Technology **50**(1-4), 297–313 (2010)
14. Shin, J.H., Jun, H.B.: On condition based maintenance policy. Journal of Computational Design and Engineering **2**(2), 119–127 (2015)
15. Si, X.S., Wang, W., Hu, C.H., Zhou, D.H.: Remaining useful life estimation–a review on the statistical data driven approaches. European journal of operational research **213**(1), 1–14 (2011)
16. Susto, G.A., Schirru, A., Pampuri, S., McLoone, S., Beghi, A.: Machine learning for predictive maintenance: A multiple classifier approach. IEEE Transactions on Industrial Informatics **11**(3), 812–820 (2014)

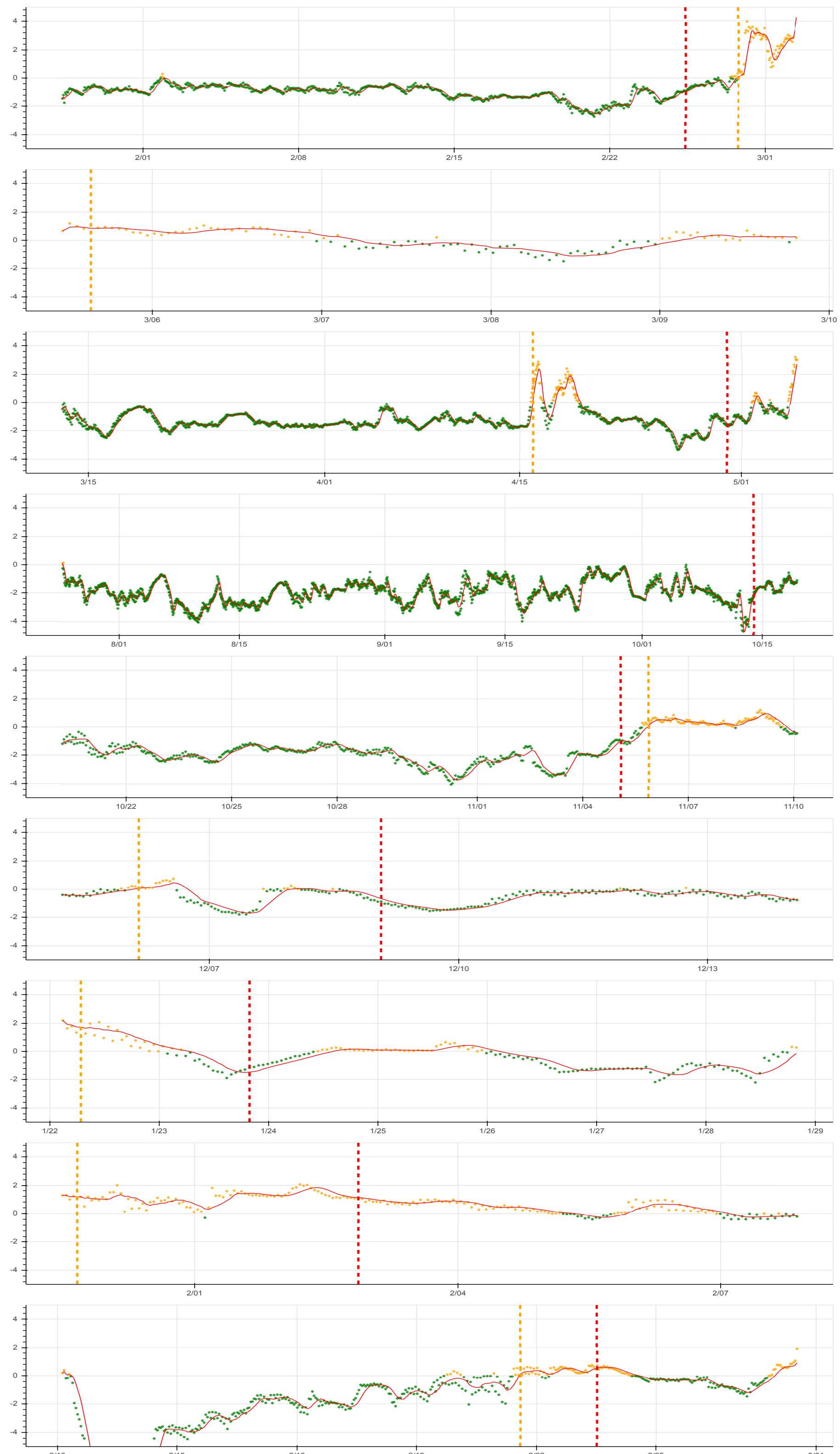

Fig. 5: Results per fold: Green points are point classified as nominal, Orange points are critical points, red solid line is the rolling average of the SVM output for eight hours, vertical dashed red line mark 5 days before failure and the vertical dashed orange line mark the alarm sent by the decision maker. On the x-axis we have the time and on the y-axis, the decision function value of the SVM.