CEN WORKSHOP AGREEMENT

CWA 17492

January 2020

ICS 35.240.50

English version

Predictive control and maintenance of data intensive industrial processes

This CEN Workshop Agreement has been drafted and approved by a Workshop of representatives of interested parties, the constitution of which is indicated in the foreword of this Workshop Agreement.

The formal process followed by the Workshop in the development of this Workshop Agreement has been endorsed by the National Members of CEN but neither the National Members of CEN nor the CEN-CENELEC Management Centre can be held accountable for the technical content of this CEN Workshop Agreement or possible conflicts with standards or legislation.

This CEN Workshop Agreement can in no way be held as being an official standard developed by CEN and its Members.

This CEN Workshop Agreement is publicly available as a reference document from the CEN Members National Standard Bodies.

CEN and CENELEC members are the national standards bodies and national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and United Kingdom.

EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brussels

Ref. No.:CWA 17492:2020 E

Contents

Page

European foreword

CWA 17492 was developed in accordance with CEN-CENELEC Guide 29 "CEN/CENELEC Workshop Agreements – The way to rapid agreement" and with the relevant provisions of CEN/CENELEC Internal Regulations – Part 2. It was agreed on 2019-09-24 in a Workshop by representatives of interested parties, approved and supported by CEN and CENELEC following a public call for participation made on 2019-04-04. It does not necessarily reflect the views of all stakeholders that might have an interest in its subject matter.

This CWA has been developed under the Task 8.3 – "Standardization" of the MONSOON (MOdel based coNtrol framework for Site-wide OptimizatiON of data-intensive processes) project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 723650 (call H2020-IND-CE-2016-17).

The final text of CWA 17492 was submitted to CEN for publication on 2019-11-18. It was developed and approved by:

- Claudio Pastrone, LINKS Foundation – Chairman
- José Antonio Jiménez Caballero, UNE – Secretary
- André Augé, Rio Tinto (France)
- Dimosthenis Ioannidis, Information Technologies Institute, Center for Research and Technology (Greece)
- Thanasis Vafeiadis, Information Technologies Institute, Center for Research and Technology (Greece)
- Vincent Bonnivard, Probayes (France)
- Bilal Azennoud, Probayes (France)
- Rosaria Rossini, LINKS Foundation (Italy)

It is possible that some elements of CWA 17492 may be subject to patent rights. The CEN-CENELEC policy on patent rights is set out in CEN-CENELEC Guide 8 "Guidelines for Implementation of the Common IPR Policy on Patents (and other statutory intellectual property rights based on inventions)". CEN and CENELEC shall not be held responsible for identifying any or all such patent rights.

The Workshop participants have made every effort to ensure the reliability and accuracy of the technical and non-technical content of CWA 17492, but this does not guarantee, either explicitly or implicitly, its correctness. Users of CWA 17492 should be aware that neither the Workshop participants, nor CEN and CENELEC can be held liable for damages or losses of any kind whatsoever which may arise from its application. Users of CWA 17492 do so on their own responsibility and at their own risk.

Comments or suggestions from the users of the CEN-CENELEC Workshop Agreement are welcome and should be addresses to the CEN-CENELEC Management Centre.

Introduction

Process industry is characterized by intense use of raw resources and energy, thus providing a context where even small optimizations can lead to high absolute savings both in terms of economic and environmental costs, if they can prove to offer predictable and replicable results. Predictive modelling techniques can be especially effective in optimizing processes in such context, but their application is not straightforward for several reasons including, e.g. the high cost of integrating large numbers of new sensors or actuators into legacy production equipment, intrinsic difficulties in monitoring physical parameters in harsh conditions, interoperability issues among existing IT systems in use, difficulties in monitoring data-intensive processes in a scalable fashion, difficulties in fusing and correlating information collected at different SCADA levels, challenges in defining and computing meaningful KPIs to ease decision-making, etc. Therefore, the deployment of model-based predictive functions in such production environments at a sustainable cost or with enough reliability is not always feasible, resulting in optimization potentials remaining untapped.

In past markets characterized by lower international competition, stable demand, relatively low labour cost and high abundance of raw materials, industry was able to remain viable just through progressive improvements in production technology, organization and logistics. The change in global competition and resources availability calls instead for a drastic re-invention and re-design of production processes and sites. In other types of production environment which are more flexible by nature, new sites can be devised which take into consideration such challenges by design. This is however not possible in capital intensive process industries, where initial investments for new production sites are prohibitive. For this reason, enabling benefits by integrating innovations in the installed process base is a fundamental step to help process industries transitioning from the current model oriented to the production of goods by consuming resources, to newer “circular” models. In this perspective, resource, cost and environmental sustainability is considered, monitored and optimized at all times, resulting in benefits for industries and society as a whole.

1 Scope

This document contains a methodology detailing the machine/deep learning techniques that should be employed, through the different steps to be followed, with the aim to predict industrial processes or equipment drifts and trigger alarms and potentially help to improve overall equipment effectiveness or the workshop performances.

NOTE The triggered alarms are related to the process in such a way a small deviation affecting the production can be detected in advance, but these alarms are not related to safety.

This document can be used as a guide by:

— Manufacturing plant managers: it contains two examples of real use cases that show the possibilities offered by machine/deep learning techniques applied to the control and optimization of manufacturing processes and to the predictive maintenance of plant machinery;

— Data Scientists: The actual use cases shown reflect the problems they will face when applying these techniques in an industrial environment, which has its own characteristics.

2 Machine/Deep learning for data-intensive industrial process

2.1 Machine/Deep learning techniques

Machine learning and Deep learning techniques are a set of methods from the field of Artificial Intelligence, which combine statistics, algorithms, and computer science. They are used to build mathematical models from sets of data, and are applied for a wide variety of tasks, such as speech recognition, image recognition, fraud detection, or product recommendations.

Those models need to be **trained**, i.e. their parameters need to be adjusted, on a so-called training dataset. Machine/deep learning techniques can be divided in two main families:

— Supervised learning, for which the data are “labeled”, i.e. the outputs of the task being modeled are known for those data;

— Unsupervised learning, with unlabeled data, where the algorithm will learn the underlying structure of the dataset.

The main algorithms used in machine learning are the following:

— Linear/Logistic Regression

— Classification and Regression Trees

— Ensemble methods

— Naive Bayes

— K-Nearest Neighbors

— K-Means Clustering

— Support Vector Machines

— Trend analysis

— Neural Networks

Deep learning is a specific family of machine learning that employs deep neural networks, e.g. neural networks with a significant number (i.e. tens) of layers.

2.2 Tasks in the context of data-intensive industrial process

In the context of data-intensive industrial processes, two main data science tasks have been considered during the MONSOON project: the predictive control and the predictive maintenance.

— Predictive control aims at giving, in real-time, recommendations to the plant end-users so as to optimize their process. The recommendations are built from a predictive machine learning model. They must be understandable by the end-user, which adds constraints on the complexity of the features being used by the models. A certain level of interpretability of the machine learning model is also necessary.

— Predictive maintenance's goal is to raise alerts when a failure of a key equipment of the plant is anticipated by a machine learning model. It is a forecasting task. Giving explanations on why the model detects such failure is important, but not always mandatory, depending on the business case.

2.3 Methodology for application in industries

The standard way of applying data science techniques to industrial business cases is the CRISP-DM (Cross Industry Standard Process for Data Mining) methodology. It is divided in six steps, as shown in Figure 1:

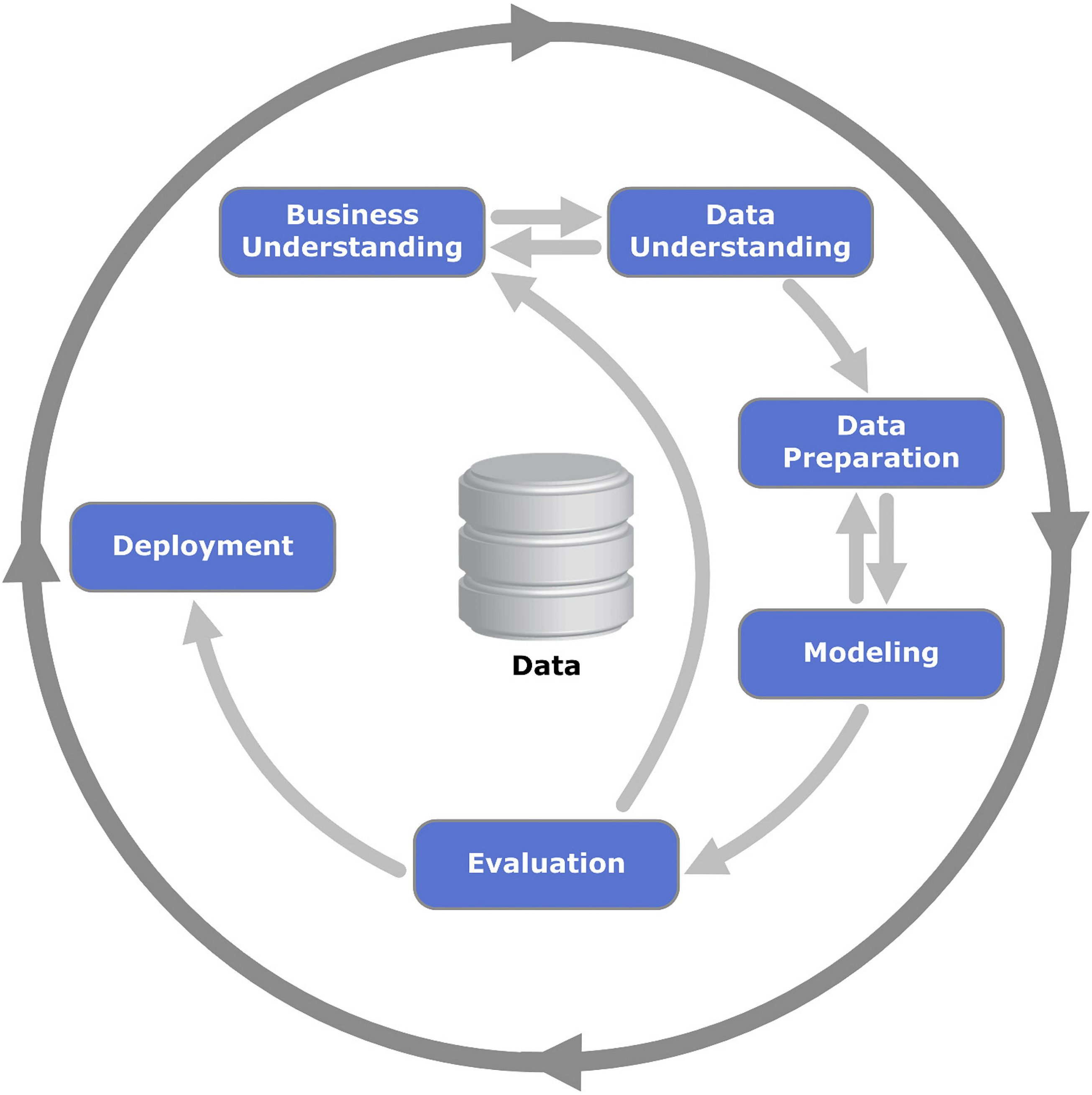


Figure 1 — CRISP-DM Methodology

1. Business Understanding

Focuses on understanding the project objectives and requirements from a business perspective, and then converting this knowledge into a data science problem definition and a preliminary plan.

2. Data Understanding

Starts with an initial data collection and proceeds with activities in order to get familiar with the data, to identify data quality problems, and to discover first insights into the data.

3. Data Preparation

The data preparation phase covers all activities to construct the final dataset from the initial raw data. Missing values replacement, anomalies removal, and data alignment are examples of tasks belonging to this step.

4. Modeling

Modeling techniques (from machine learning algorithms) are selected and applied. Since some techniques like neural networks have specific requirements regarding the form of the data, there can be a loop back here to the data preparation.

5. Evaluation

Once one or more models have been built that appear to have high quality based on whichever key performance indicators have been selected, these need to be tested to ensure they generalize against unseen data and that all key business issues have been sufficiently considered. The end result is the selection of the champion model(s).

6. Deployment

Generally, this will mean deploying a code representation of the model into an operating system to score or categorize new unseen data as it arises and to create a mechanism for the use of that new information in the solution of the original business problem. Importantly, the code representation must also include all the data preparation steps leading up to modelling so that the model will treat new raw data in the same manner as during model development.

3 Example of application – Anode quality prediction for aluminium production

3.1 Use case description and scope

Anodes quality, and particularly their density, constitute a key component to the aluminium electrolysis reaction. It impacts directly the quantity and the quality of the produced aluminium. Anodes are produced by the paste plant. The anodes production is highly controlled and the process leads mainly to high quality anodes. Nevertheless, the paste plant goes through some special periods where the anodes quality could be improved. Understanding the root causes of these lower quality periods constitutes a challenging task but a key component for maintaining a high-quality production.

A machine learning model for monitoring the anode quality and understanding the process causes behind the decreasing quality periods was developed. A recommendation module has also been developed in order to recommend the best process parameters changes for maintaining a high production quality rate. The next parts will describe thoroughly the whole pipeline for modelling the anode density using process data and machine learning. The last part will cover the recommendation methodology.

3.2 Presentation of process parameters

Paste plant sensors provide a rich source of data. The sensors continuously measure hundreds of signals at a one second frequency. For the modelling task, 41 process signals resuming information have been chosen. They consist of measurements of intensities, powers, temperatures, speeds, rates of flow of raw material, etc. The list of parameters to use was selected after several iterations with process experts from the plant. All those parameters are stored in a storage system, for which a dedicated connector was developed in the context of the MONSOON project, in order to export the data to MONSOON's Datalab and Runtime Container platforms.

The paste plant produces anodes continuously, with the paste going through different equipment before being used to form the anodes. Therefore, at a given anode production time, it is needed to take into account the transit time of the paste in the plant for associating the signals sensors. However, this transit time is not precisely known. Therefore, entire periods of anodes production have been chosen, instead of individual anodes. The selected period duration is 30 minutes.

3.3 Data preparation for modelling

The primary data partition is of 30 minutes. Each shift of 8 hours has been divided into sixteen 30 minute periods.

The pipeline of data preparation could be summarized in 4 phases:

a) Detection of paste plant stops periods.

b) Annotation of periods density quality: ‘good’ or ‘improvable’.

c) Linking process data to each 30 minute period.

d) Computing features summarizing each process parameter during the 30 minute period.

3.3.1 Detection of paste plant stops

During the stoppages and the restarting of the paste plant, the sensors signals are known to show abnormal behaviours. Moreover, these stops are known to cause a variability of anodes density. The aim for this analysis is to detect the causes of non-optimal anode quality during **standard** running of the paste plant, thus periods containing paste plant stoppages are ignored.

Three rules were defined by the process experts to detect a stop period:

— INT-MOY-MALAXEUR < 200

— Or VIT-MOT-VIS-DEMANDEE < 50

— Or DEB-INSTANTANE-DOSEUR <= 1

These three signals measure respectively the intensity of the paste mixer, the dosimeter speed, and the material flow, thus allowing to know the status (working/stop) of the paste plant. The thresholds were chosen so that this definition remains valid in time.

Due to signals instability during the following periods, the following has been also discarded:

— Periods shorter than 24 minutes between two stops.

— 5 minutes before each stop.

— 10 minutes after each stop, or 1 hour if the stop lasted more than 30 minutes.

An illustration of the behaviour of these variables for an 8 h period in which two paste plant stoppages occur is given in Figure 2.

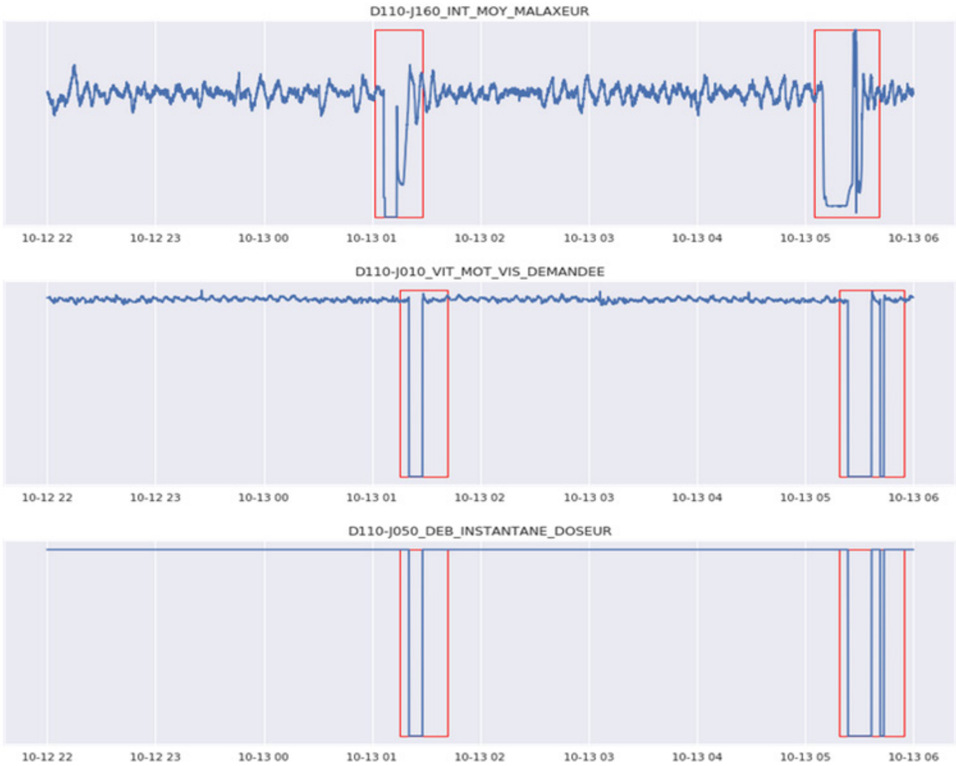


Figure 2 — Example of signals used for identifying paste plant stops – The periods in red are identified as stops

Figure 3 shows the anodes density produced during the same shift. The drop-in anode density during the stop's periods can be seen.

Figure 3 — Anode density during a shift containing two stops

3.3.2 Annotation of anode quality: 'good' or 'improvable'

Each period of 30 minutes is annotated as "good" or "improvable". A threshold is defined on the median density of all the anodes produced during the period. Periods above the threshold are annotated as "good".

The value of the threshold depends on the training sample used for training the model (this part will be detailed in the section "Re-training of the model" below). Basically, all the periods density used for the training have been taken and the 5 % percentile of the median density have been selected as a threshold to decide if the period is "improvable" or "good": periods with median density below the threshold are tagged as "improvable".

3.3.3 Linking process data to each 30 minute period

It is necessary to associate each sensor signal to its corresponding anode production period. Indeed, the production of anode paste is a continuous process, with the paste going through different machines before being used to form the anodes. Therefore, at a given anode production time, it is necessary to consider the transit time of the paste in the plant for associating the signals sensors to the produced anode. For instance, as it takes ~10 min for the paste to go from the mixing to the forming chains, the measurements related to the paste mixer must be associated to the anodes produced ~10 min later. In practice, the delays which need to be added to each sensor have been estimated by the plant process experts.

3.3.4 Features creation for the model

In order to resume the signal of each process parameters during the 30 minute periods, two simple statistical features are computed: median and the standard variation. Note that these features are particularly simple in order to allow the **interpretability** of the model's decision, a requirement for being able to give recommendation on process parameters.

Let's write $\boldsymbol{y_i}$ the quality type of anodes produced in a given $\boldsymbol{i}$ period:

$$\boldsymbol{y_i} = \begin{cases} 0\left('\mathit{improvable}'\mathit{quality}\right) \\ 1\left('\mathit{good}'\mathit{quality}\right) \end{cases}$$

We note $\boldsymbol{Y} = \left(\boldsymbol{y_i}\right)_{1 \le \boldsymbol{i} \le \boldsymbol{n}}$, the vector of quality types for the *n* periods. Modelling the anode period quality ('good' or 'improvable') using features of the process parameters is characterized as a classification task.

It is necessary to compute for each signal during the 30 minute period some features that summarize its behavior during the period.

We note $\boldsymbol{X} = \left(\boldsymbol{x_{i,j}}\right)_{1 \le \boldsymbol{i}, \boldsymbol{j} \le \boldsymbol{n}, \boldsymbol{p}}$, the training samples where $\boldsymbol{x_{i,j}}$ are the computed features during the period *i* of the signal *j*.

For this analysis, *p=41* sensors signals have been selected. The total number of features created for each period is 82. The idea is to identify the quality of the periods, ***Y***, based on the training samples, ***X***, using a machine learning model. The model will predict a probability of belonging to the 'improvable-quality' class, and a decision threshold will be chosen on the probabilities above which the periods will be considered as 'improvable-quality'.

Note that special treatments have been conducted for some signals, mainly vibrocompactors signals which alternate between down and up states (Figure 4). For those signals, only the upward periods which describe the state of the anode paste in the vibrocompactor have been considered.

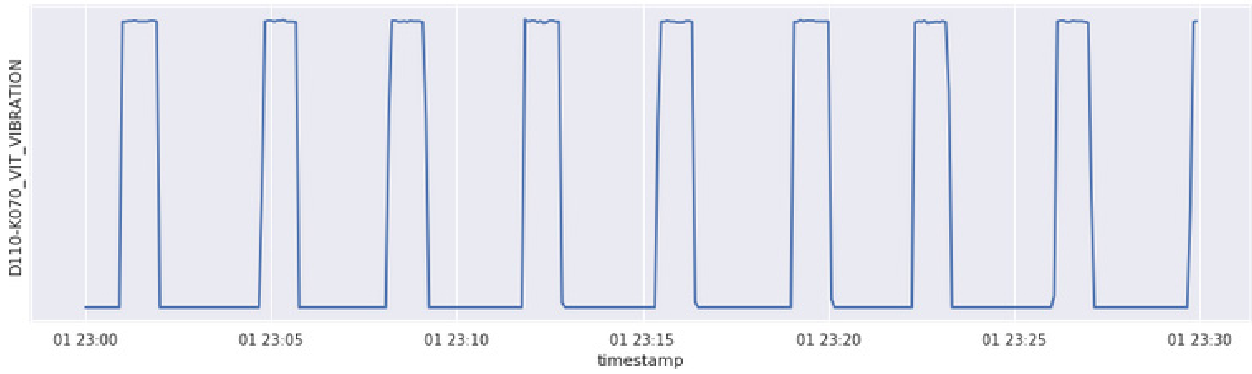


Figure 4 — Example of a vibrocompactor signal alternating between down and up states

3.4 Modelling the anode quality and the re-training of the model

3.4.1 Presentation of the model

It is wanted to assign a class to each period by modelling the probability of a period being improvable as a function of the features created from the process parameters. XGBoost's Gradient Boosted Decisions Trees [4] as a classifier have been selected for the modelling. It builds an ensemble of weak classifiers (small decision trees here) in an iterative way. This model belongs to the category of ensemble learning models and is more robust to class imbalance, correlation of features, and overfitting of the data.

3.4.1.1 Training of the model

The next graph explains the whole pipeline for model re-training. After each shift (8 hours), the following has to be done:

— Select the past months (maximum 6 months, minimum 3 months) as a training data.

— Split training data to 30 minute periods.

— Compute features (median and standard-deviation) on the 30 minute periods.

— Annotate each period as “good” or “improvable”: the threshold is the 5 % percentile of the median density during the training period (6 months).

— Predict the periods quality on the periods of the next shift.

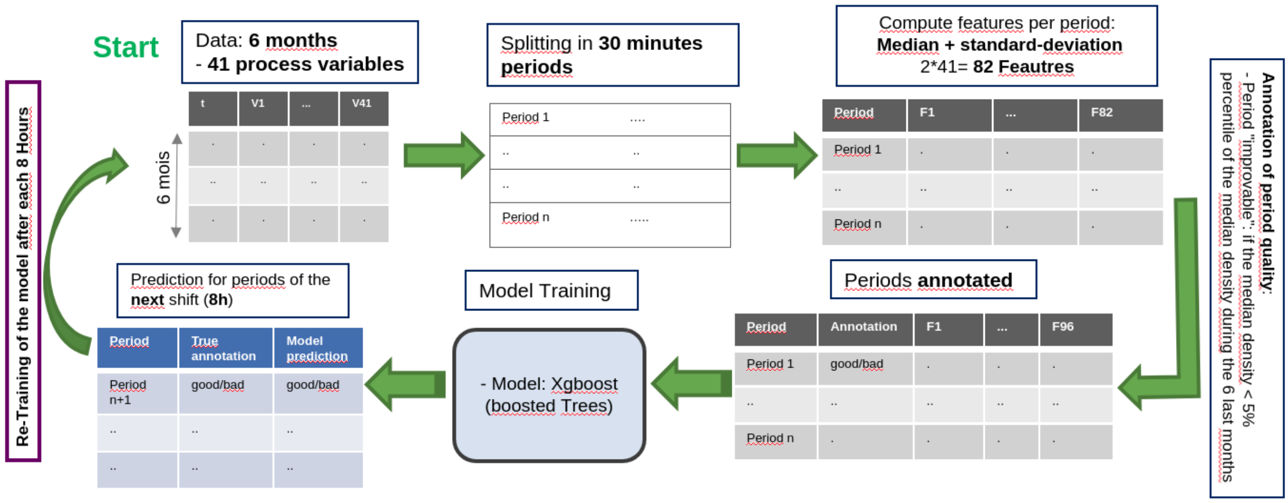


Figure 5 — Pipeline of anode quality model re-training

The latest dataset used for the modelling starts from April 2018 and ends in March 2019. The ratio of classes “improvable”/”good” is of the order of 8 %. The challenge of the class imbalance was handled by using a tree-based ensemble method, XGBoost, as well as by tuning its hyperparameters. The tuning was done during the first iteration of the project, by applying a grid-search method over a 3-folds cross-validation.

3.4.1.2 Measuring the performances of the model

As explained on the re-training part, at a given time *t*, the previous months have been selected (between [3 months, 6 months]) for training the model and the next 8 hours as a test set for the prediction. The window is then slid by 8 hours (up to *t* + *8h*), the model is retrained, and tested on the following 8 hours. This cross-validation method is better suited in the context of time series.

In order to overcome the problem of misleading performance estimation using accuracy in the context of imbalanced classes, more appropriate estimators have been used. Precision and Recall for the ‘improvable quality’ class as performance indicators have been employed.

The next Figure resumes the performances of the models tested for the period July 2018 – March 2019.

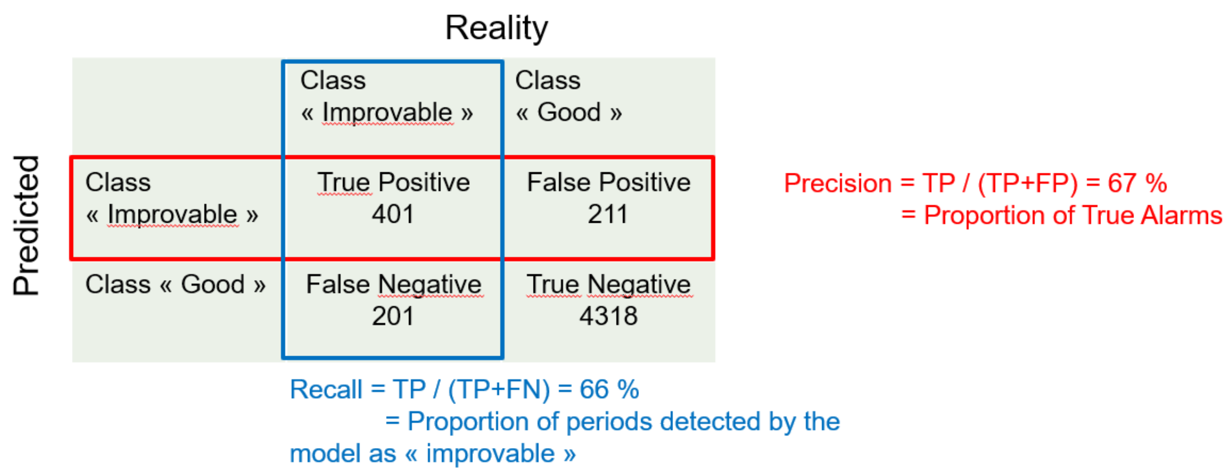


Figure 6 — Model performances: Confusion matrix, Precision and Recall

The models show good performances as it detects 66 % of “improvable” periods (Recall) with a 67 % Precision. In comparison, a random model, who would randomly pick 8 % of the periods and attribute them to the “improvable” class, would have statistically both a Precision and Recall of 8 %.

Next figure represents the Receiver Operating Curve (ROC). It is another way of measuring the model performance, by showing the true positive rate versus the false positive rate, for different decision thresholds on the probabilities predicted by the model. The higher is the space over the diagonal line (which represents the performances of a random model), the better is the model.

The Area Under Curve AUC is of 0.93, demonstrating a good prediction capability of the models.

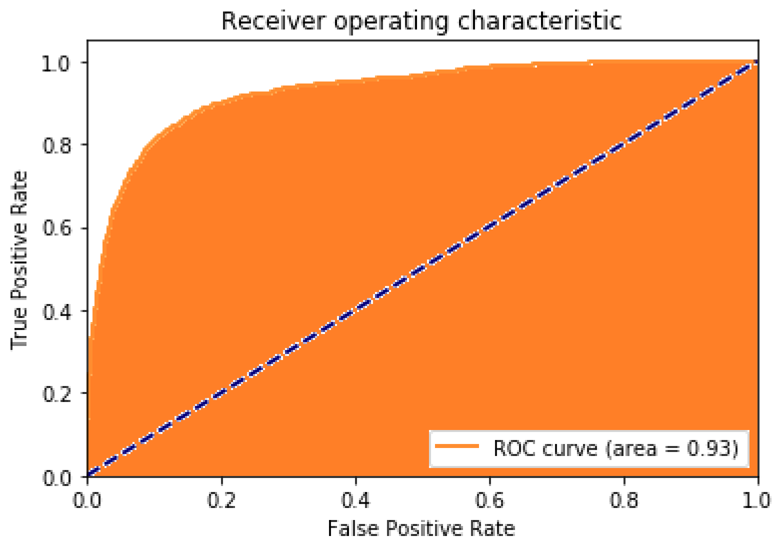


Figure 7 — Model performance: ROC curve

3.5 Understanding the root causes of decreasing anode quality: Model explicability

The idea behind the previous modelling is to model the anodes quality as a function of the process parameters. Training ensemble models for this task leads to better prediction performances as compared to simple learners but in counterpart the interpretability of the model is lost.

Domain experts need to understand the decision of the model in order to trust it and to help them discovering other unseen causes of the decision. There are many methods developed in order to overcome the challenge of model explicability. SHAP has been chosen (Shapley Additive exPlanations) [5,6] for the task as it has been proven in literature as the most robust method to date for this purpose.

SHAP module relies on SHAPLEY values which were introduced in Game Theory. The idea is to model the model predictions as a weighted sum of each feature-value contribution. Basically, for each individual instance and the value of each feature, the SHAPLEY value gives the contribution of that feature-value on the predicted value by the model.

In practice, the SHAP module is used after each true prediction of the model for the ‘improvable’ class. It is possible to obtain the list of the five features that contributed the most to the probability of being an ‘improvable’ period, in an online mode, i.e. after each new prediction of the model. Figure 8 shows four examples of SHAP’s interpretations, as shown in the Grafana [7] interface built for the anode quality function in the Runtime Container.

Bad Quality Periods explanations

Time	Period ID	Explanation 1	Explanation 2	Explanation 3	Explanation 4
2019-02-01 08:00:00	20190201M8H0	D110-M120_DEPRESSION_FILTRE__standard_deviation	D110-J173_TEMP_EAU_REFROIDIS__quantile__q_0.5	D110-K060_VIT_VIBRATION__quantile__q_0.5	D110-J173_DEB_EAU_REFROIDISSEMENT_J173__quantile__q_0.5
2019-02-01 08:30:00	20190201M8H1	D110-J173_TEMP_EAU_REFROIDIS__quantile__q_0.5	D110-M120_DEPRESSION_FILTRE__standard_deviation	D110-K060_VIT_VIBRATION__quantile__q_0.5	D110-K070_VIT_VIBRATION__quantile__q_0.5
2019-02-01 09:00:00	20190201M9H0	D110-J173_TEMP_EAU_REFROIDIS__quantile__q_0.5	D110-M120_DEPRESSION_FILTRE__standard_deviation	D110-M120_DEPRESSION_FILTRE__standard_deviation	D110-J030_VIT_MOT_VIS_DEMANDEE__quantile__q_0.5
2019-02-01 09:30:00	20190201M9H1	D110-M120_DEPRESSION_FILTRE__standard_deviation	D110-J173_TEMP_EAU_REFROIDIS__quantile__q_0.5	D110-K060_VIT_VIBRATION__quantile__q_0.5	D110-J030_VIT_MOT_VIS_DEMANDEE__standard_deviation

Figure 8 — Table resuming the 3 (for illustration purpose) most important explainers of decreasing anode quality for some 'improvable' periods

The proposed five explainers will give insights to the experts on which process parameters to visualize in order to investigate the causes of the decreasing quality of anodes periods.

Figure 9 illustrates the behaviour of some of the process parameters proposed by SHAP as possible causes of decreasing anodes quality for some 'improvable' periods (red colour), as displayed in the Grafana interface from the Runtime Container.

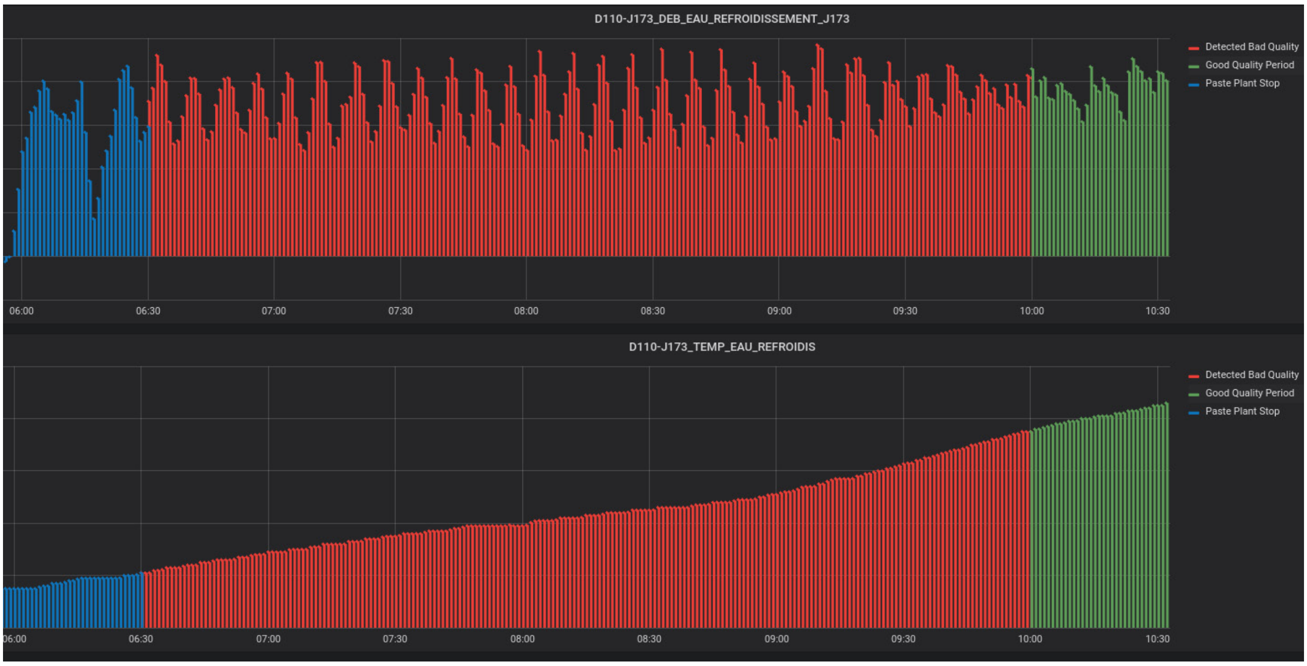


Figure 9 — Visualization of some process variables proposed by the explainer module as possible cause for the decreasing quality periods

3.6 Recommendation module for maintaining highly quality anodes density

After detecting an 'improvable' period with the classifier model, in addition to the 'explanations' given by SHAP, a recommendation is given on which actionable parameters should be modified, and how, in order to increase the anode density.

Indeed, the 'explanations' given by SHAP are a list of features from the entire list of process parameters used as input by the XGBoost classifier. However, most of these parameters are **sensors**, and cannot be modified easily by the plant team.

However, among them, a list of **actionable parameters** has been identified by the plant team. The aim is therefore to propose a recommendation on **three actionable parameters to modify together** in order to increase the density, when an 'improvable' period is detected. Three parameters have been chosen to end up with recommendations that are easy to implement by the plant team.

Figure 10 illustrates the principle of the approach. The period **p0** is below the low-density threshold (in red) and is therefore considered as an 'improvable' period. If the process conditions stay the same for the next periods, the density will stay low (dashed lines). However, if the actionable parameters are changed according to the function's recommendation, the density should increase (green lines).

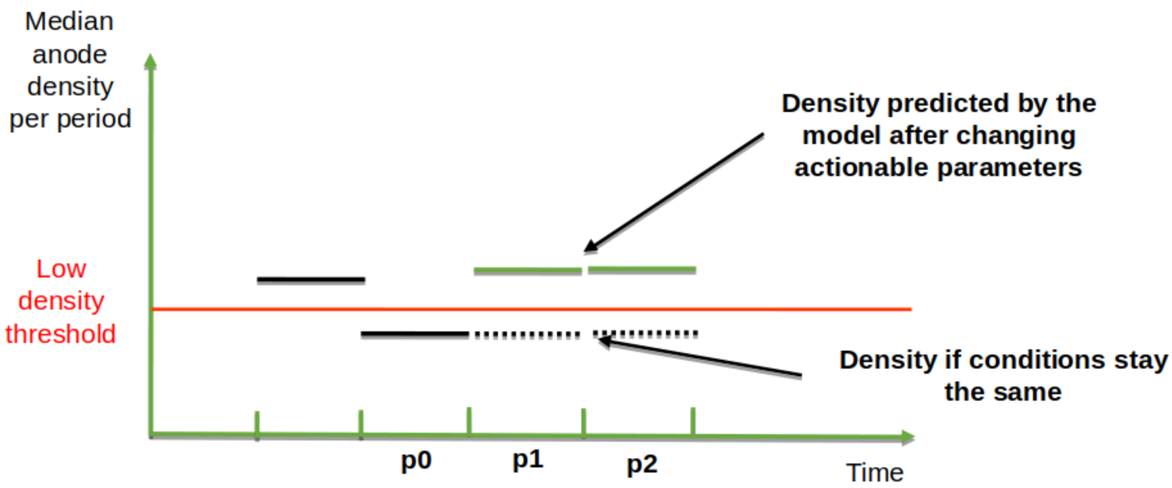


Figure 10 — Simulation for the recommendation module after an 'improvable-density' period

To be able to provide the recommendations, a simulation function has been developed. The function is a machine learning regressor, which is trained to predict the median density of a 30-minutes period, given the process parameters. An XGBoost regressor has been chosen with the same training strategy as the previous classifier model but using only the median features as variables for the model (the recommendations will be given on the median values of the process parameters, and not the standard deviations, which are more difficult to act on).

Figure 11 shows the performances of the regressor, for testing periods between July 2018 and March 2019. The X axis shows the actual median densities, and the Y axis the predicted ones. The model has a Root Mean Squared Error of 2.85 (the order of magnitude of the density is typically 1640). The model is performing well, except for a handful of very low-density periods, for which the regressor predicts higher densities. This is probably due to the very low number of examples of these low densities.

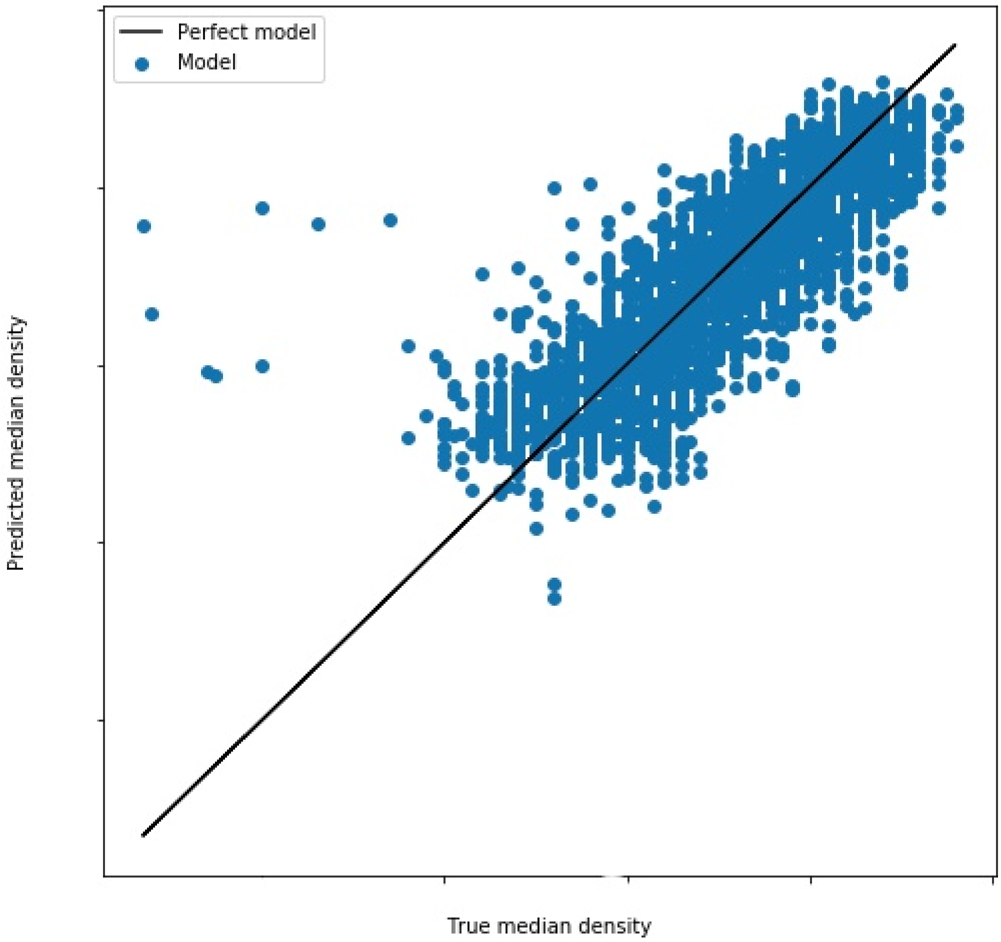


Figure 11 — Regressor model predictions vs actual median densities

Figure 12 illustrates the entire pipeline for executing the recommendation module. The pipeline is the following:

1. For a period of 'improvable' density which was detected as such by the anode quality classifier, the process started by scanning all the triplets of actionable parameters in their respective range. The range is deduced from the last 6 months of available data. For each triplet of actionable parameters values, the regression module is applied to simulate the expected median density.

   However, when testing a triplet of actionable parameters, it is necessary to take into account their possible **correlations with other parameters**: if the actionable parameter A1 is highly correlated with sensor parameter S5, then increasing A1's value without increasing S5's value is not physically realistic. To mitigate this issue, the multiple correlation coefficients are computed between each triplet of actionable parameters, and each other process parameter. The computation is done by training a linear regression model between the triplet of actionable parameters and the given other parameter, and computing the square root of the coefficient of determination. The training is done on the last 6 months of available data, and the outliers are being removed with a 3-sigma filtering. If the correlation coefficient is above 0.3 for a given process parameter, then when scanning the triplet of actionable parameters values, the correlated process is also modified according to the regression function.

2. The triplet of actionable parameters that lead to the highest increase of median density, according to the simulation model, is selected.

3. The recommendation is displayed, consisting of the list of actionable parameters to change, and the associate values to reach.

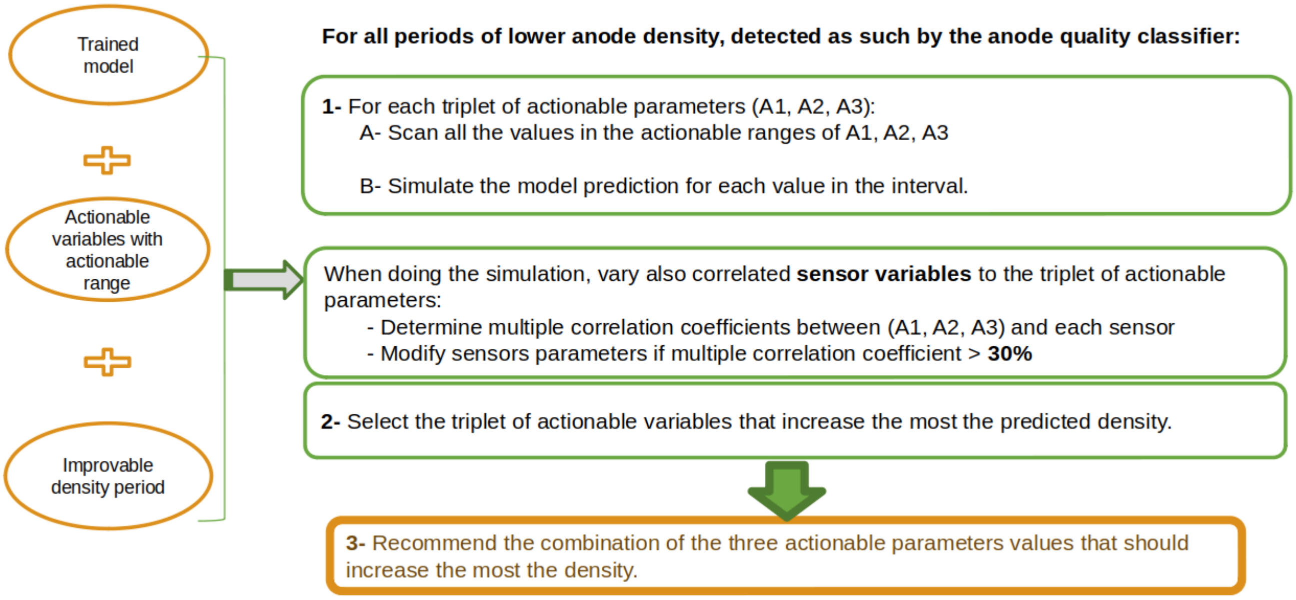
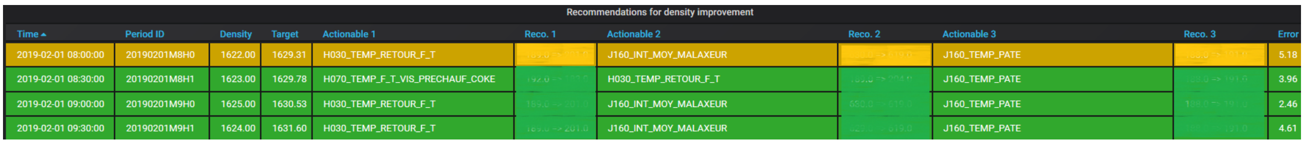


Figure 12 — Detailed pipeline for recommendation

In order to provide a confidence level of the recommendation, the error between the actual median density of the period of interest, and the one obtained with the regression module is computed. If the error is larger than 5 (in the unit of the density), the recommendation is flagged as "less reliable". This threshold was chosen as such, as it is the typical order of magnitude of density improvement obtained with the recommendation module. In the dedicated Grafana user interface, the recommendations are displayed in a table (see Figure 13), where the 'less reliable' ones appear in yellow, while the other are in green.



Recommendations for density improvement

Time	Period ID	Density	Target	Actionable 1	Reco. 1	Actionable 2	Reco. 2	Actionable 3	Reco. 3	Error
2019-02-01 08:00:00	20190201M8H0	1622.00	1629.31	H030_TEMP_RETOUR_F_T	[illegible]	J160_INT_MOY_MALAXEUR	[illegible]	J160_TEMP_PATE	[illegible]	5.18
2019-02-01 08:30:00	20190201M8H1	1623.00	1629.78	H070_TEMP_F_T_VIS_PRECHAUF_COKE	[illegible]	H030_TEMP_RETOUR_F_T	[illegible]	J160_TEMP_PATE	[illegible]	3.96
2019-02-01 09:00:00	20190201M9H0	1625.00	1630.53	H030_TEMP_RETOUR_F_T	[illegible]	J160_INT_MOY_MALAXEUR	[illegible]	J160_TEMP_PATE	[illegible]	2.46
2019-02-01 09:30:00	20190201M9H1	1624.00	1631.60	H030_TEMP_RETOUR_F_T	[illegible]	J160_INT_MOY_MALAXEUR	[illegible]	J160_TEMP_PATE	[illegible]	4.61

Figure 13 — Presentation of the recommendation module output – The actual recommendations were blurred for confidentiality purpose

3.7 Deployment

The function has been deployed to Runtime Container installed in the plant. The function is split in two main modules: the training module, and the execution module.

— The training module automatically re-trains all the models necessary for the execution module (i.e. the classification model, the regression model, and the linear regression model). The training is launched every 8 hours, one hour before the starting of the next shift (i.e. at 5 A.M., 1 P.M., and 9 P.M.). The module reads the data of the last 6 months in the database available in the Runtime Container; pre-processes the data and compute the features; trains all the machine learning models; and finally save them in a dedicated folder, for the execution module.

— The execution module applies the function to the latest 30-minute period available. It is executed every 30 minutes. The module queries the raw data in the database; pre-processes them; selects the latest trained models; and applies the function. For each period, it predicts the anode quality status (i.e. normal or 'improvable'); and in case the period was 'improvable' and detected as such by the classifier, it generates the list of five 'explanations' and generates the triplet of recommendations for improving the density. The results are exported to the database for visualization purpose.

A specific user interface has been developed for the Runtime Container in the plant. It consists of several dashboards made with Grafana. The dashboards allow the end-user to visualize all the process parameters being used by the model; and shows the 'explanations' and recommendations proposed by the function. A specific dashboard allows the monitoring of the performances of the function, in terms of Precision and Recall.

4 Example of application – Prediction of breakdowns in industrial equipment

4.1 General

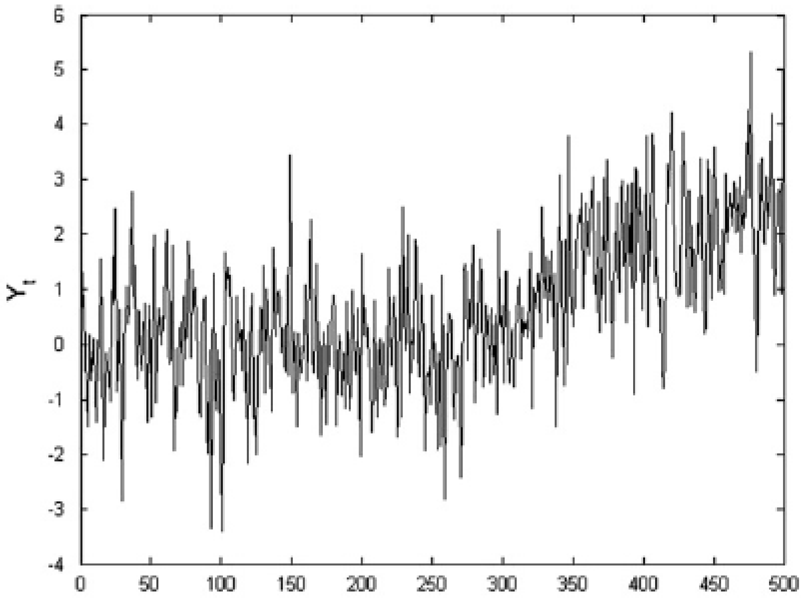
This clause shows an example of techniques and methodologies for trend analysis for predictive maintenance of industrial equipment, specifically an industrial mixer. With the implementation of such techniques it is expected to provide detection of possible deviations from normal conditions, based on the needs of industrial processes.

Generally, the use of trend analysis provides a robust signal processing technique on the real time incident detection problem. This clause contains an application of the Slope Statistic Profile function. Slope Statistic Profile function tests the fault scenario where the linear trend of the time series has downward deviations from no trend situations. The method can be easily modified to work on other hypothetical use cases like the detection of upward changes or the detections of both upward and downward changes.

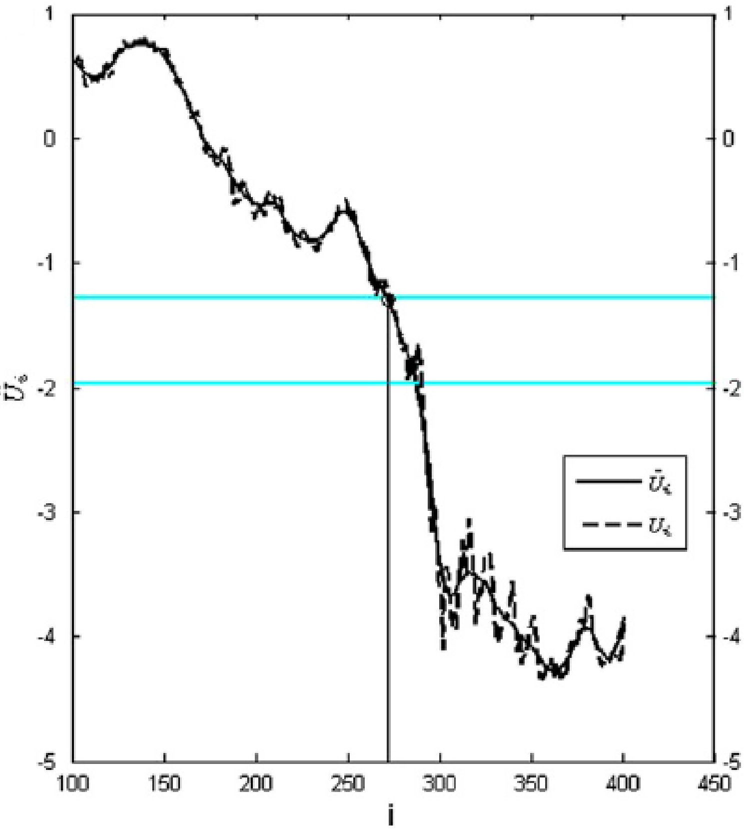
4.2 Slope Statistic Profile (SSP)

Slope Statistic Profile, denoted hereinafter as *SSP*, is a method that detects the single structural break *T*, denoted hereafter as incident, in a time series using a standard parametric linear trend test, denoted hereinafter as *t*-statistic. The *t*-statistic is calculated on overlapping sliding data windows of size *w* with sliding step one, along the time series. In this way the profile of the *t*-statistic is obtained, denoted as $\{\tilde{U}_i\}$, for $i = 1 + \left[\frac{w}{2}\right], \ldots, n - \left[\frac{w}{2}\right]$, where $[x]$ is the integer part of *x*. The form of this profile depends on time series characteristics, i.e. the strength of the autocorrelation, the distribution of the residuals, the strength of the linear trend, as well as, the size of the sliding data window *w*.

The profile of the *t*-statistic $\{\tilde{U}_i\}$, exhibits small fluctuations (glitches) due to edge effects of the local data windows and therefore the profile curve is smoothed using a zero-phase filter of a small order, set to about $5\ \%$ of *w*. Such a small filter order removes the glitches in $\{\tilde{U}_i\}$, but maintains its original signature. The smoothed value of $\tilde{U}_i$ is denoted as U_i and refer to as *U*-profile. In a short presentation of the method below, the situation from no trend to a positive trend will be assumed, as shown in Figure 14 (a). Other types of change between no trend and trend can be treated similarly.



(a) Time series of length n = 500 with an onset of linear trend with coefficient β = 0.01 at T = 250 and residuals generated by $AR(1)$ with coefficient φ = 0.16 and normal input white noise



(b) The profile of *t*-statistic $\{\tilde{U}_i\}$ using local window of size w = 200 and the filtered profile $\{U_i\}$, as denoted in the legend. The horizontal lines denote the thresholds $-t_{w-2,1-0.975}$ and $-t_{w-2,1-0.90}$ and the vertical line the estimated $\hat{T}$ at time 274

Figure 14

The t-statistic for the parametric linear trend test is $t = \hat{\beta} / s(\hat{\beta}) \sim t_{w-2}$, where $\hat{\beta}$ the trend parameters and $s(\hat{\beta})$ is the standard error of $\hat{\beta}$. The null hypothesis of no trend is rejected at the significance level a if $|t| \geq t_{w-2,1-a/2}$.

A first candidate for the breakpoint T is the time point at which the profile crosses the threshold line of rejection of the null hypothesis of no trend at $\pm t_{w-2,1-a/2}$, where a is the significance level, w is the size of the sliding window and t follows the Student distribution with *w-2* degrees of freedom [3]. One could consider one-sided test and use a smaller threshold magnitude $t_{w-2,1-a}$ if there is knowledge for the direction of the linear trend (-$t_{w-2,1-a}$ for positive trend and $t_{w-2,1-a}$ for negative trend). The U-profile most probably does not exhibit a sudden change from small magnitudes, in the absence of trend, to large magnitudes, in the presence of trend, but there is a rather smooth transition due to the use of sliding windows with step one. This consideration comes along with the belief that there are not sudden and abrupt changes in natural variations. Thus, a better estimate of incident T should be searched at times regarding magnitudes of $\{U_i\}$ smaller than $t_{w-2,1-a/2}$. Figure 14 (b) shows the profile of *t*-statistic as long as the filtered profile with the two thresholds. We confine the search of T to a time interval corresponding to the profile segment bounded by $t_{w-2,1-0.975}$ and $t_{w-2,1-0.90}$, for the 0.05 and 0.20 significance levels for the two-sided test, respectively, and find the smallest magnitude of $\{U_i\}$ within this segment. The corresponding time point for this value is the estimated incident $\hat{T}$. For the example of Figure 14 (b) the estimated incident $\hat{T} = 274$ of the true incident $T = 250$ corresponds to the profile crossing of the lower bound at $-t_{w-2,1-0.90} = -1.286$. This example illustrates a straightforward detection with *SSP* in a well-behaving situation.

In order to both cases of upward and downward linear trend to be covered by *SSP* method, two segments are necessary to be defined. Thus, for positive linear trend the $(t_{w-2,1-a_1/2}, t_{w-2,1-a_2/2})$ segment is defined and will be denoted hereafter as upper segment. The bounds of upper segment ,$t_{w-2,1-a_1/2}$, $t_{w-2,1-a_2/2}$ will be denoted hereafter as UB_1 and UB_2, respectively. For negative linear trend the segment $(-t_{w-2,1-a_2/2}, -t_{w-2,1-a_1/2})$ is defined and will be denoted hereafter as lower segment. The bounds of lower segment, $-t_{w-2,1-a_1/2}$, $-t_{w-2,1-a_2/2}$ will be denoted hereafter as LB_1 and LB_2, respectively. The significance levels a_1 and a_2 for two side test are set 0.20 and 0.05, respectively [1].

At this point, a brief description of the linear trend test statistic that is used in *SSP* method is given. In the following, the parametric linear trend test for a sliding window of size w on the time series $\mathrm{Y_t}$, $\mathrm{t} = 1, \ldots, \mathrm{n}$, is presented. Thus, for the first window $[\mathrm{Y_1}, \ldots, \mathrm{Y_w}]^\mathrm{T}$ the least square estimator for the trend parameter β is obtained as

$$\hat{\beta} = \frac{\sum_{t-1}^{w}\left(t-\bar{t}\right)Y_t}{\sum_{t-1}^{w}\left(t-\bar{t}\right)^2} \tag{1}$$

where $\bar{t}$ is the average time. The standard error of $\hat{\beta}$ can be estimated with several approaches. Here, the best two approaches are presented [2]: the autocovariance and the power spectrum approach.

In autocovariance approach, the estimated standard error of $\hat{\beta}$ is calculated by

$$s_1\left(\hat{\beta}\right)=\left\{c\left[\gamma_0+2c\sum_{s=2}^{w}\sum_{t=1}^{s-1}\left(t-\bar{t}\right)\left(s-\bar{t}\right)\gamma_{s-t}\right]\right\}^{1/2} \tag{2}$$

where $c=\frac{12}{w\left(w^2-1\right)}$. In (2), γ_k is replaced with the respective estimate of $\hat{\gamma}_\kappa$, except at $k=0$ where $w\hat{\gamma}_0/\left(w-2\right)$ is used in order to estimate γ_0 [3]. Thus, the estimated standard error of $\hat{\beta}, s_1\left(\hat{\beta}\right)$ is derived.

In power spectrum approach, the estimated standard error of $\hat{\beta}$ is calculated by

$$s_2\left(\hat{\beta}\right)=\left[2\int_0^{0.5}W\left(f\right)S\left(f\right)\right]^{1/2} \tag{3}$$

where $W\left(f\right)=\left|\sum_{t=1}^{w}b_t e^{-2\pi if}\right|^2$ with $b_t=\frac{t-\bar{t}}{\sum_{t=1}^{w}\left(t-\bar{t}\right)^2}$ and $S\left(f\right)$ denotes the sample power spectrum of ε_t given as $S\left(f\right)=\frac{1}{2\pi}\left(\hat{\gamma}_0+2\sum_{k=1}^{w-1}\hat{\gamma}_\kappa\cos(2\pi fk\right)$. $\hat{\gamma}_\kappa$ denotes the estimate of the kth order autocovariance of ε_t, given as $\hat{\gamma}_\kappa=\frac{1}{w}\sum_{t=1}^{w-k}\hat{\varepsilon}_{t+k}\hat{\varepsilon}_t$ for $k>0$, where $\hat{\varepsilon}_t=Y_t-\hat{a}-\hat{\beta}t$ are the estimated residuals ($\hat{a}=\overline{Y}_t-\hat{\beta}t$ and $\overline{Y}_t$ is the mean of the time series), and $\hat{\gamma}_0=\frac{1}{w-2}\sum_{t=1}^{w}\hat{\varepsilon}_t^2$ for $k=0$. Thus, the estimated standard error of $\hat{\beta}, s_2\left(\hat{\beta}\right)$ is derived. The t-statistic for the parametric linear trend test is $t=\hat{\beta}\big/s_k\left(\hat{\beta}\right)$, where $k=1$ is the autocovariance approach and $k=2$ the power spectrum approach.

Both approaches of standard error estimation have different characteristics that affect the t-statistic. The autocovariance approach is more sensitive to small changes in linear trend and that makes the $t=\hat{\beta}\big/s_1\left(\hat{\beta}\right)$ test statistic more condescending while the power spectrum approach gives high test power to $t=\hat{\beta}\big/s_2\left(\hat{\beta}\right)$ compared to other test statistics for both correlated and white noise residuals.

SSP methodology has the ability to detect all the kinds of changes on linear trend of a time series (i.e. significant upward changes, upward changes, significant downward changes and downward changes).

4.3 Application to the prediction of breakdowns in industrial equipment

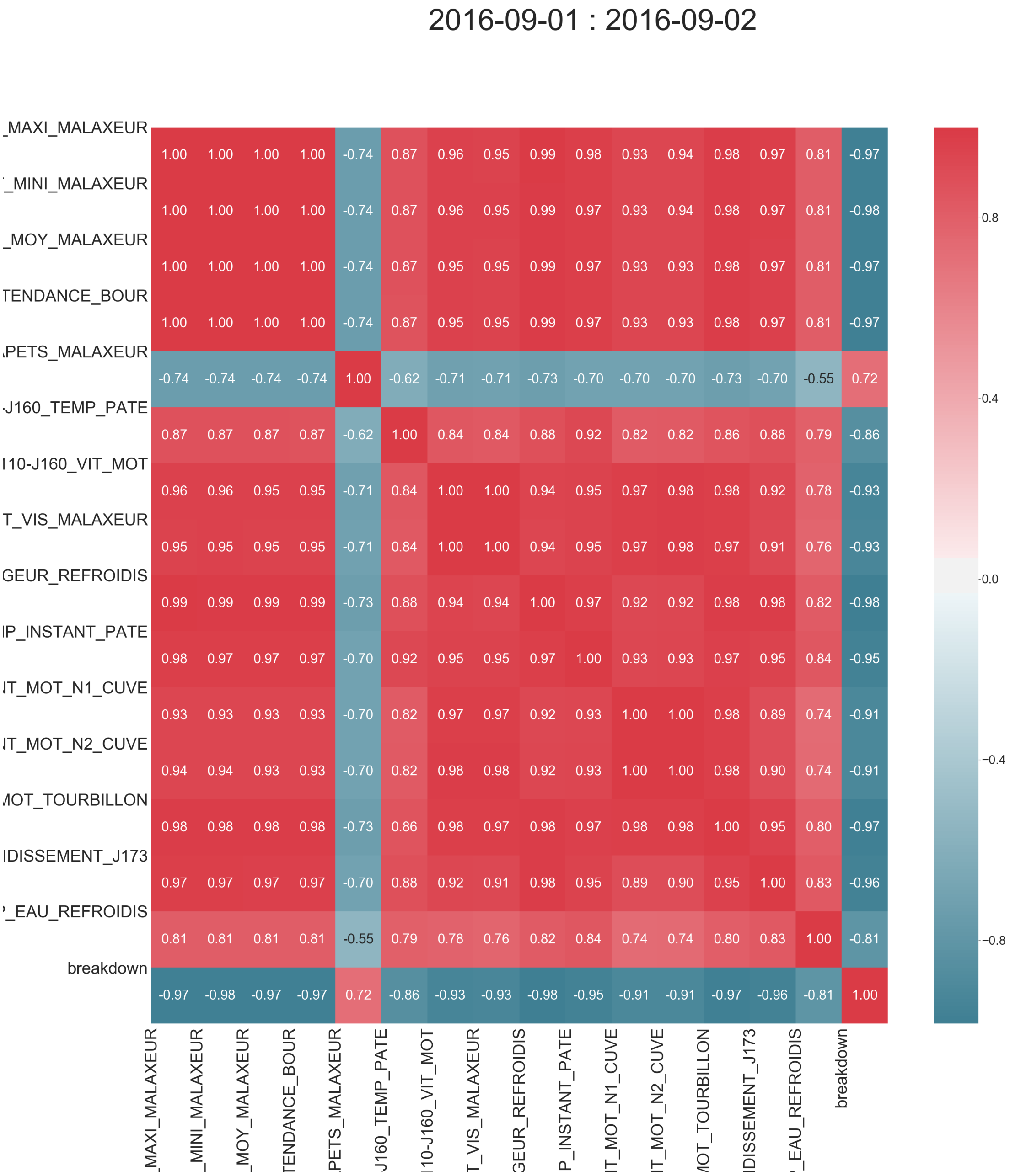
4.3.1 SSP methodology application

SSP methodology has been applied on the data of an industrial mixer and more specifically on the variables given in the Table 1 below:

Table 1 — Mixer variables names

1	MIXER VARIABLES NAMES
2	D110-J170_NIVEAU_MELANGEUR_REFROIDIS
3	D110-J160_INT_MAXI_MALAXEUR
4	D110-J160_INT_MINI_MALAXEUR
5	D110-J160_INT_MOY_MALAXEUR
6	D110-J160_MES_CORRIGE_TENDANCE_BOUR
7	D110-J160_MES_OUV_CLAPETS_MALAXEUR
8	D110-J160_PUISSANCE_MOY_MALAXEUR
9	D110-J160_TEMP_PATE
10	D110-J160_VIT_VIS_MALAXEUR
11	D110-J160_VIT_MOT

A correlation analysis based on *Point-Biserial* methodology took place for the variables of Table 1 for a random selection of a subset of the total breakdown occurrences. Such a subset can be seen in Figure 15 where correlations between variables and breakdowns can be observed for periods 2016-09-01 to 2016-09-02 and 2018-11-08 12:40:00 to 2018-11-10 20:13:10. Variables with higher correlation coefficient were tested against SSP methodology with sliding windows of 30, 60 and 90 elements. The data for each variable were recorded from September 2016 until March 2019. The overall recording consisted of 7971008 entries in a 12 second time interval. The SSP methodology was set to detect the significant upward and downward changes of the selected variables because these changes point to faults according to the evaluation method.


2016-09-01 : 2016-09-02
_MAXI_MALAXEUR
_MINI_MALAXEUR
_MOY_MALAXEUR
TENDANCE_BOUR
PETS_MALAXEUR
J160_TEMP_PATE
10-J160_VIT_MOT
T_VIS_MALAXEUR
GEUR_REFROIDIS
P_INSTANT_PATE
T_MOT_N1_CUVE
T_MOT_N2_CUVE
IOT_TOURBILLON
IDISSEMENT_J173
_EAU_REFROIDIS
breakdown
1.00 1.00 1.00 1.00 -0.74 0.87 0.96 0.95 0.99 0.98 0.93 0.94 0.98 0.97 0.81 -0.97
1.00 1.00 1.00 1.00 -0.74 0.87 0.96 0.95 0.99 0.97 0.93 0.94 0.98 0.97 0.81 -0.98
1.00 1.00 1.00 1.00 -0.74 0.87 0.95 0.95 0.99 0.97 0.93 0.93 0.98 0.97 0.81 -0.97
1.00 1.00 1.00 1.00 -0.74 0.87 0.95 0.95 0.99 0.97 0.93 0.93 0.98 0.97 0.81 -0.97
-0.74 -0.74 -0.74 -0.74 1.00 -0.62 -0.71 -0.71 -0.73 -0.70 -0.70 -0.70 -0.73 -0.70 -0.55 0.72
0.87 0.87 0.87 0.87 -0.62 1.00 0.84 0.84 0.88 0.92 0.82 0.82 0.86 0.88 0.79 -0.86
0.96 0.96 0.95 0.95 -0.71 0.84 1.00 1.00 0.94 0.95 0.97 0.98 0.98 0.92 0.78 -0.93
0.95 0.95 0.95 0.95 -0.71 0.84 1.00 1.00 0.94 0.95 0.97 0.98 0.97 0.91 0.76 -0.93
0.99 0.99 0.99 0.99 -0.73 0.88 0.94 0.94 1.00 0.97 0.92 0.92 0.98 0.98 0.82 -0.98
0.98 0.97 0.97 0.97 -0.70 0.92 0.95 0.95 0.97 1.00 0.93 0.93 0.97 0.95 0.84 -0.95
0.93 0.93 0.93 0.93 -0.70 0.82 0.97 0.97 0.92 0.93 1.00 1.00 0.98 0.89 0.74 -0.91
0.94 0.94 0.93 0.93 -0.70 0.82 0.98 0.98 0.92 0.93 1.00 1.00 0.98 0.90 0.74 -0.91
0.98 0.98 0.98 0.98 -0.73 0.86 0.98 0.97 0.98 0.97 0.98 0.98 1.00 0.95 0.80 -0.97
0.97 0.97 0.97 0.97 -0.70 0.88 0.92 0.91 0.98 0.95 0.89 0.90 0.95 1.00 0.83 -0.96
0.81 0.81 0.81 0.81 -0.55 0.79 0.78 0.76 0.82 0.84 0.74 0.74 0.80 0.83 1.00 -0.81
-0.97 -0.98 -0.97 -0.97 0.72 -0.86 -0.93 -0.93 -0.98 -0.95 -0.91 -0.91 -0.97 -0.96 -0.81 1.00
0.8
0.4
0.0
−0.4
−0.8

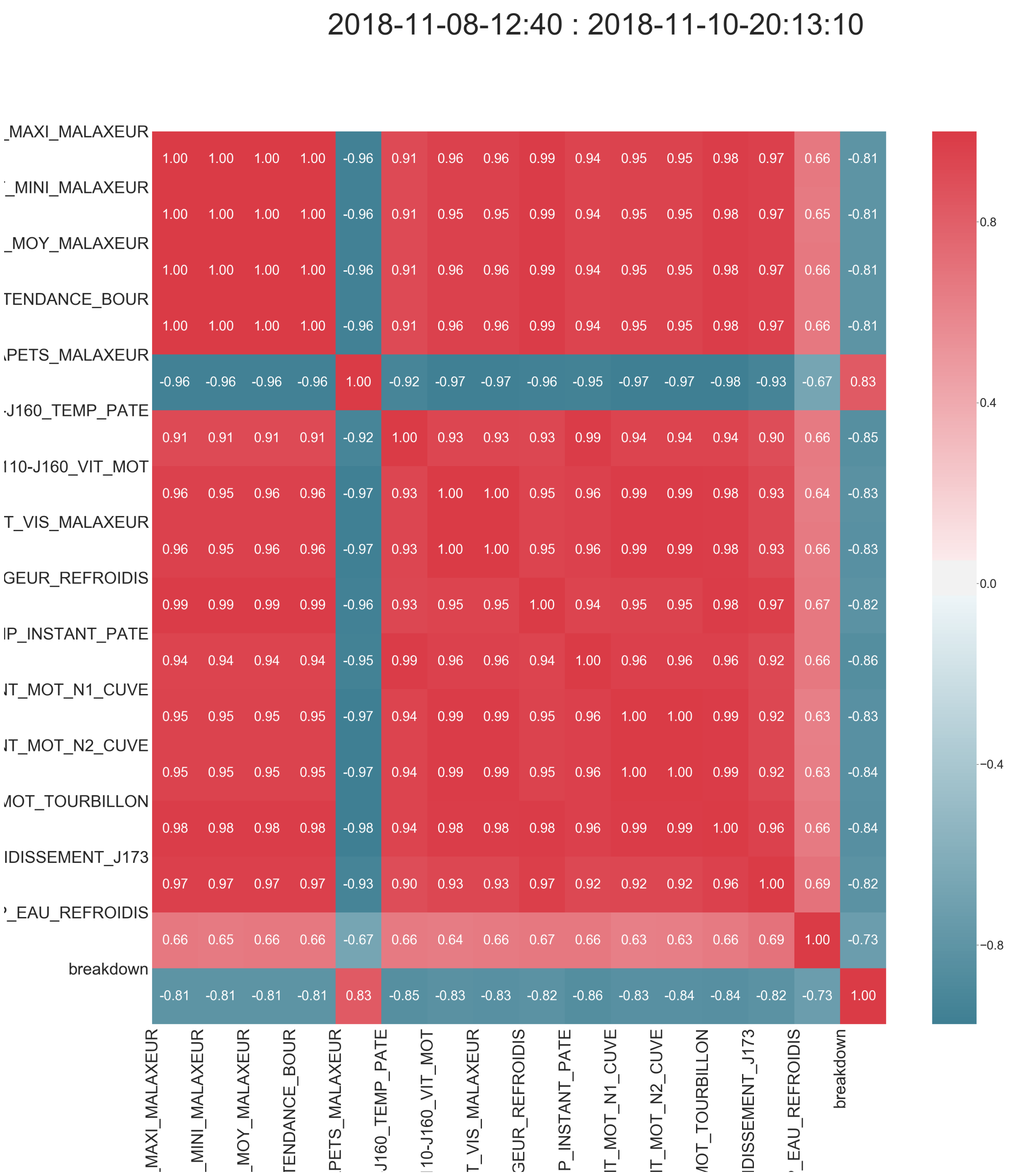


Figure 15 — Correlation matrices of mixer variables for periods 2016-09-01 to 2016-09-02 and 2018-11-08 12:40:00 to 2018-11-10 20:13:10

Trend analysis methodology was also modelled as a binary classification problem that predicts whether or not a new breakdown will occur in the forthcoming minutes. More specifically, at each moment the result of the SSP from the past 30, 60 and 90 elements will signal or not a possible malfunction that will happen in the next 3, 5, 10, 15 or 20 minutes. For each execution of the algorithm was chosen as training set a slice of the recorded data for a specific mixer variable. The size of SSP's sliding window and a threshold that directly affects the upward and downward detection bounds of the algorithm were given as parameters. The results of the execution are saved on a csv file for evaluation. The file consists of three columns that represent the timestamp, prediction and the actual breakdown. For each time step the algorithm will mark 1 in the prediction column, if a significant upward or downward trend is caught and 0 otherwise. An example of the file structure is presented on Figure 16. When SSP finishes running, the csv file is created, and the results can be evaluated. The evaluation technique is described in detail in subclause 4.3.2.

Timestamp	Prediction	Breakdown
2016-12-31 12:04:50	1	0
2016-12-31 12:05:00	1	0
2016-12-31 12:05:10	1	0
2016-12-31 12:05:20	1	0
2016-12-31 12:05:30	1	0
2016-12-31 12:05:40	1	0
2016-12-31 12:05:50	1	0
2016-12-31 12:06:00	1	1
2016-12-31 12:06:10	1	1
2016-12-31 12:06:20	1	1

Figure 16 — SSP's execution resulting file – Describes a forthcoming prediction and the actual time of a breakdown

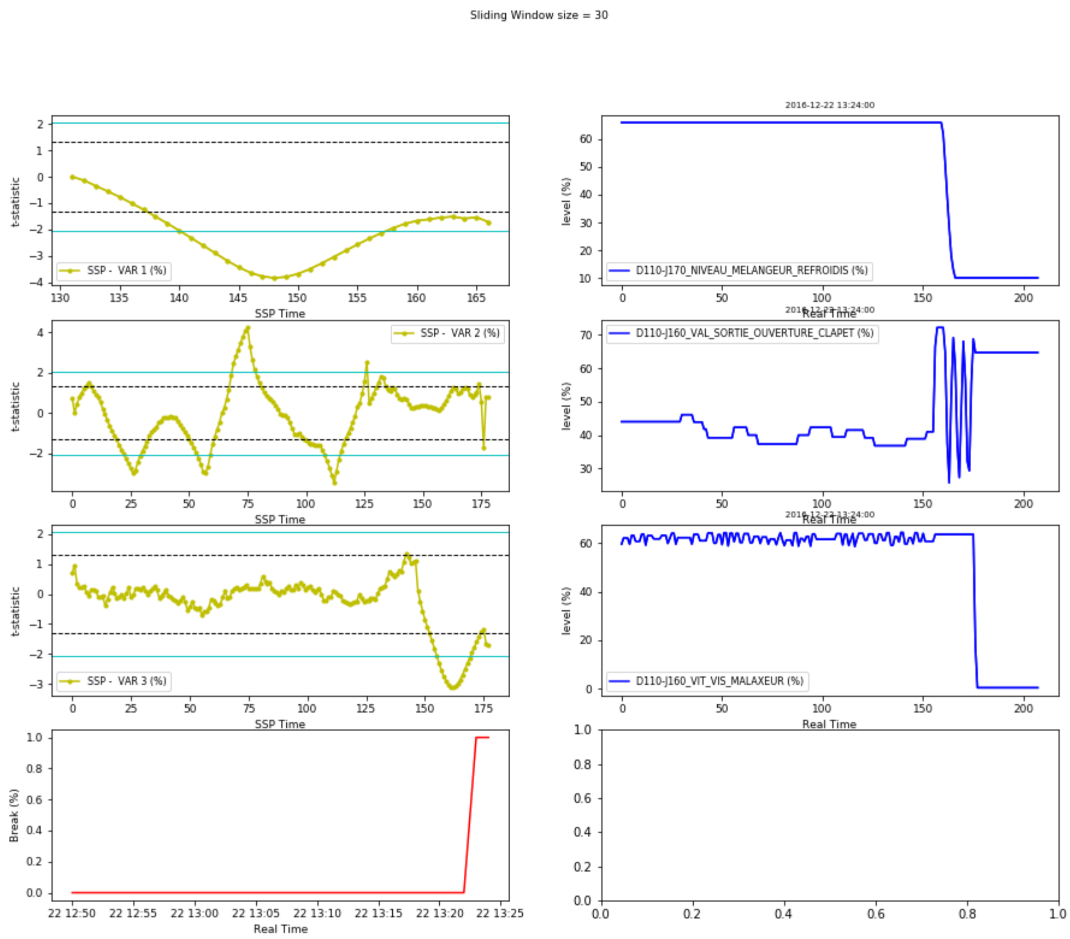


Figure 17 — The 3 top right figures show the real values of 3 variables while the 3 to left show their SSP values. The bottom left figure shows the time a breakdown occurred

An instance of a malfunction detection from 22-12-2016 12:50 to 22-12-2016 13:25 is shown in Figure 17. The three right plots (blue color) show the original values of the variables D110-J170_NIVEAU_MELANGEUR_REFROIDIS, D110-J160_MES_OUV_CLAPETS_MALAXEUR and D110-J160_VIT_VIS_MALAXEUR from top to bottom, respectively. Their corresponding SSP values are shown on the top three left plots. The fourth bottom left plot shows the time of the breakdown (value=1). The first variable which is also the one with the smallest false positive rate, signals a possible malfunction at 13:18:00, 5 minutes before the actual breakdown happens. For each variable, there is a signal for a malfunction when the SSP value is smaller or equal to -2 or, equal or greater than 2. The SSP application for the second variable shows a lot of signals for the forthcoming breakdown while the SSP for the third variable shows one signal just before the occurrence of the breakdown.

4.3.2 Evaluation Algorithm

The evaluation algorithm handles the malfunction detection problem as a time-series classification prediction model where at each point in time, given a variable, it predicts the existence or not of a malfunction in the near future. It takes as a parameter the time window in which a prediction can be made. The algorithm starts by iterating over the csv with the historical data and for each entry (that is represented by a triplet [Timestamp, Prediction, Breakdown]), it checks the current Prediction value and compares it with all the Breakdown values of the next timestamps until timestamp is equal to an upper prediction bound (time window). If the prediction is 1 and there exists at least one 1 in the breakdown column in the given time window, we have a correct prediction (true positive). If the prediction is 1 and in the given time window exist only zeros, we have a false prediction (false positive). If the prediction is 0 and in the time window exist only zeros we have a correct prediction (true negatives) and if there exists at least one 1 we have again a false prediction (false negatives). For instance, if for evaluation has been selected a time window of 20 minutes, the scope of the algorithm is to predict if a breakdown will happen in the forthcoming 20 minutes at most. The evaluation finishes with a sum of all the true positives, false positives, true negatives and false negatives. From them can be derived common evaluation metrics such as accuracy, precision and recall. True positive represent correct alarm for a forthcoming or existing malfunction. False positive represent a false alarm. True negatives mean that in the forthcoming minutes will not appear a malfunction and indeed does not appear. False negatives mean the model falsely predicts no malfunction. Figure 18 shows the SSP predictions and the actual breakdowns from 22-12-2016 to 9-1-2017 using the variable D110-J160_VIT_VIS_MALAXEUR with sliding window equal to 90 samples. Green lines denote a prediction for a forthcoming malfunction, while the red dots denote the presence of a breakdown.

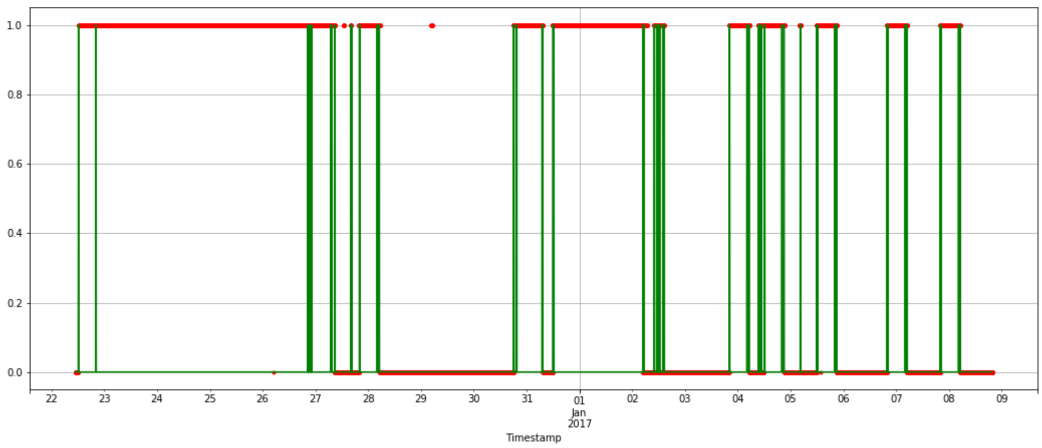


Figure 18 — SSP predictions (green) and machine breakdowns (red) using D110-J160_VIT_VIS_MALAXEUR for 22-12-2016 to 9-1-2017

4.3.3 Results

The evaluation technique described in subclause 4.3.2 run on SSP results of 6000000 entries from 22-12-2016 to 10-11-2018. For each variable, sliding windows of 30, 60 and 90 elements were given as parameters to the SSP algorithm. The evaluation algorithm also had been executed for various prediction window sizes (10, 20, and 30 minutes). The variable with the best prediction results is D110-J160_VIT_VIS_MALAXEUR and its results are provided in Table 2 in percentages (%). The reset of the variable results are shown in Table 3 through Table 6. The great number of true negatives can explain high accuracy. For instance, for the variable of Table 2, for the time window of 20 minutes there are 113680 true positives, 17398 false positives, 29228 false negatives and 2256811 true negatives.

Table 2 — D110-J160_VIT_VIS_MALAXEUR – sliding window size 90

Prediction Windows	Precision	Recall	F-Score	Accuracy
10 minutes	81 %	64 %	72 %	98 %
20 minutes	87 %	81 %	84 %	98 %
30 minutes	82 %	57 %	67 %	95 %

Table 3 — D110-J170_NIVEAU_MELANGEUR_REFROIDS – sliding window size 90

Prediction Windows	Precision	Recall	F-Score	Accuracy
10 minutes	55 %	64 %	59 %	91 %
20 minutes	60 %	74 %	66 %	93 %
30 minutes	57 %	45 %	50 %	91 %

Table 4 — D110-J160_MES_OUV_CLAPETS_MALAXEUR – sliding window size 90

Prediction Windows	Precision	Recall	F-Score	Accuracy
10 minutes	30 %	52 %	38 %	84 %
20 minutes	35 %	66 %	46 %	88 %
30 minutes	33 %	61 %	43 %	85 %

Table 5 — D110-J160_PUISSANCE_MOY_MALAXEUR – sliding window size 90

Prediction Windows	Precision	Recall	F-Score	Accuracy
10 minutes	44 %	51 %	47 %	83 %
20 minutes	52 %	59 %	55 %	85 %
30 minutes	41 %	52 %	46 %	83 %

Table 6 — D110-J160_MES_CORRIGE_TENDANCE_BOUR– sliding window size 90

Prediction Windows	Precision	Recall	F-Score	Accuracy
10 minutes	56 %	51 %	53 %	89 %
20 minutes	68 %	65 %	66 %	94 %
30 minutes	61 %	60 %	60 %	93 %

D110-J160_VIT_VIS_MALAXEUR is able to better capture the breakdowns in contrast to the other tested variables. An example of another variable behavior is shown in Figure 19. The SSP application on variable *D110-J170_NIVEAU_MELANGEUR_REFROIDIS* captures most of the breakdowns but also signals many false alarms.

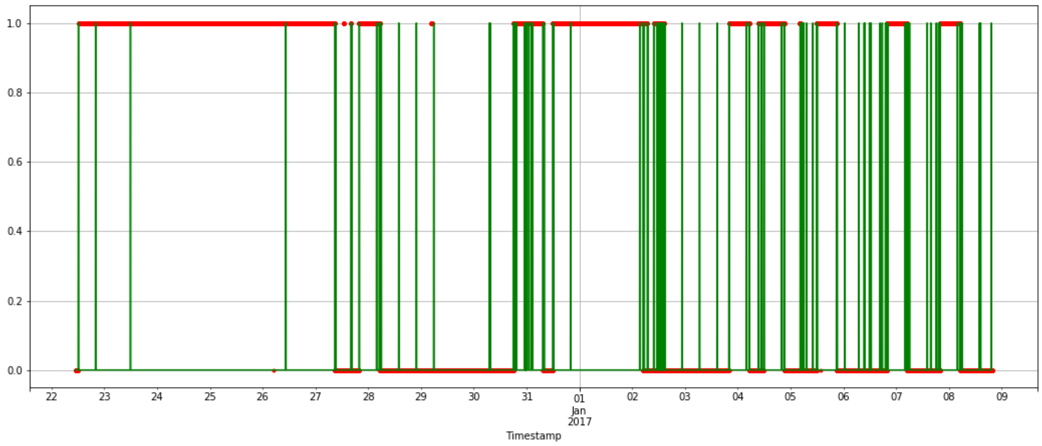


Figure 19 — SSP predictions (green) and machine breakdowns (red) using D110-J170_NIVEAU_MELANGEUR_REFROIDS for 22-12-2016 to 9-1-2017

Bibliography

[1] VAFEIADIS T., BORA-SENTA E., KUGIUMTZIS D. Estimation of linear trend onset in time series. *Simulation Modelling Practice and Theory*. 2011, **vol. 19** pp. 1384–1398

[2] VAFEIADIS T., BORA-SENTA E., KUGIUMTZIS D. Evaluation of linear trend tests using resampling techniques. *Communication in Statistics Part B: Simulation and Computation*. 2008, **vol. 37** (no. 5) pp. 907–923

[3] WOODWARD W.A., GRAY H.L. Global warming and the problem of testing for trend in time series data. *American Meteorological Society*. 1993, **vol. 6** pp. 953–962

[4] XGBOOST. http://xgboost.readthedocs.io/en/latest/

[5] LUNDBERG S., LEE S. A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems*. 2017, **vol. 30** pp. 4765–4774

[6] SHAP package in Python: https://github.com/slundberg/shap

[7] Grafana website: https://grafana.com/grafana