Validating Condition-Based Maintenance Algorithms through Simulation

Marcel Chevalier, Léo Dupont, Sylvain Marié, Frédérique Roffet, Elena Stolyarova, William Templier, Costin Vasile

Validating Condition-Based Maintenance Algorithms Through Simulation

Marcel Chevalier, Léo Dupont, Sylvain Marié, Frédérique Roffet,
Elena Stolyarova, William Templier, Costin Vasile

***Abstract*—** Industrial end users are currently facing an increasing need to reduce the risk of unexpected failures and optimize their maintenance. This calls for both short-term analysis and long-term ageing anticipation. At Schneider Electric, we tackle those two issues using both Machine Learning and First Principles models. Machine learning models are incrementally trained from normal data to predict expected values and detect statistically significant short-term deviations. Ageing models are constructed from breaking down physical systems into sub-assemblies, then determining relevant degradation modes and associating each one to the right kinetic law. Validating such anomaly detection and maintenance models is challenging, both because actual incident and ageing data is rare and distorted by human interventions, and incremental learning depends on human feedback. To overcome these difficulties, we propose to simulate physics, systems and humans – including asset maintenance operations – in order to validate the overall approaches in accelerated time and possibly choose between algorithmic alternatives.

***Keywords*—** Degradation models, Ageing, Anomaly detection, Soft Sensor, Incremental learning.

I. Introduction

NOWADAYS, digitization and Industrial Internet of Things (IIoT) make it extremely easy to collect a vast amount of data concerning electrical assets during their operational life in real time conditions on customer plants. It allows users to get information on major environmental and usage conditions for products in real situations, together with data on observed failures.

Collected data can be used to learn normal behavior models of assets. Such models are relevant to detect anomalies, characterized by a statistically significant deviation from the so-called "current normal situation". While this kind of anomaly detection method is suited for short term asset monitoring, it is not appropriate for long term degradation trends. To capture such trends, first principles degradation models can be used (i.e. physical degradation models such as Arrhenius law); they complete the monitoring system, to get a global view on asset health. Both approaches are detailed in [1].

Validating such approaches represents a significant challenge, as only little real incident data is available from the field while degradation spans across several years of normal operation. Long-term degradation data collection is impacted by actual maintenance operations that may not have happened at optimal time. Furthermore, in the Machine Learning-based approach, models are created from collected data using incremental learning strategies. During this process, useful data may be accidentally dropped, while fault data may be mistakenly injected as "normal" in the update steps.

Simulation and validation of maintenance models is first presented in section II; incremental learning-based short-term virtual sensor models are addressed in section III. We finally conclude in section IV.

II. Long-Term

In this part, we refer to a long-term approach to compute the ageing of assets with respect to time, environmental conditions, usage conditions, and maintenance operations. These ageing models can be used to define appropriate time-based maintenance.

In the following sections we will present how these models can be used to simulate alternate system designs and maintenance scenarios, e.g. condition-based maintenance.

A. Proposed approach

Each asset to monitor is decomposed in sub-assemblies, all associated with one or several degradation modes [1]. These degradation modes act in competition for each concerned sub-assembly. At each point in time, the most impacting degradation mode is selected for each sub-assembly, and the most impacted sub-assembly is used to compute the consumed lifetime. Maintenance operations may replace the currently most impacted sub-assembly with another one, or change the current most impacting degradation mode with another one.

Fig. 1 illustrates a simulation run for a given asset, showing the Consumed lifetime (%) over time. Each spike is due to a maintenance operation of replacement of a sub-assembly,

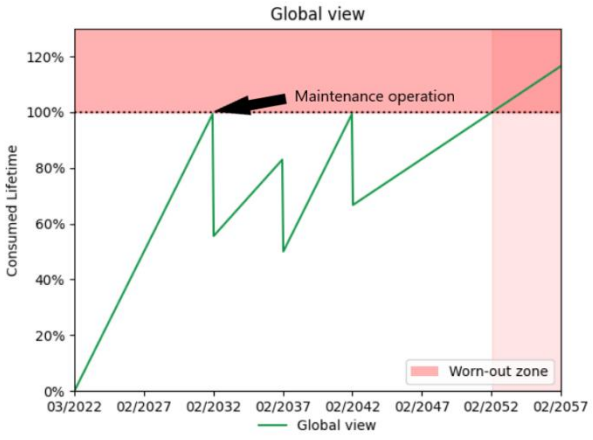

Fig. 1 Long term simulation

All authors are with Schneider Electric Industries, 160, avenue des Martyrs, 38000 Grenoble, FRANCE (e-mail: first.last@se.com).

which resets the ageing value. The first maintenance operation is done on the most impacted sub-assembly. After this event, the consumed lifetime is reset to ~60% and the degradation speed (slope) changes, as the most impacted sub-assembly is now a different one (the first one has been maintained).

What-if analysis is defined in the literature as a data-intensive simulation whose goals are to inspect the behavior of a complex system [11]. Initially focusing on scalar data, what-if scenarios nowadays embed more and more timeseries data to improve precision in the analysis [8]. Two kinds of what-if analysis can be considered:

- *Sensitivity analysis*: by generating a great number of scenarios, we can test the sensitivity of the ageing models to refine them. This analysis allows verifying and quickly modifying the models, so as to be as reliable as possible;

- *Scenario analysis*: it is also possible to generate realistic scenarios, to match reality as much as possible. Doing so, different choices can be simulated and confronted in order to find the best solution.

In this study, we focus on scenario analysis to explore two dimensions: alternate usage contexts (B) and alternate maintenance scenarios (C).

B. What-if analysis: alternate usage contexts

By modeling various scenarios, we can simulate the ageing process in a long-term view.

Our simulations of environmental conditions are based on weather simulation data [4] to provide the most realistic environmental scenarios. Naturally, these scenarios depend on the location of the customer site, together with the future usage of our products. We also have to link all the influencing factors and the customer events. For example, if a customer wants to add an air conditioner, it is possible to simulate the direct impact on environmental entries. In this case, the temperature remains constant; the humidity could be also static, but there are possible drawbacks like a raise of dust due to a new ventilation. So, with a realistic simulation, we can assess the impact of the air conditioner to advise the customer about this choice. An example of that case is provided in Fig. 2.

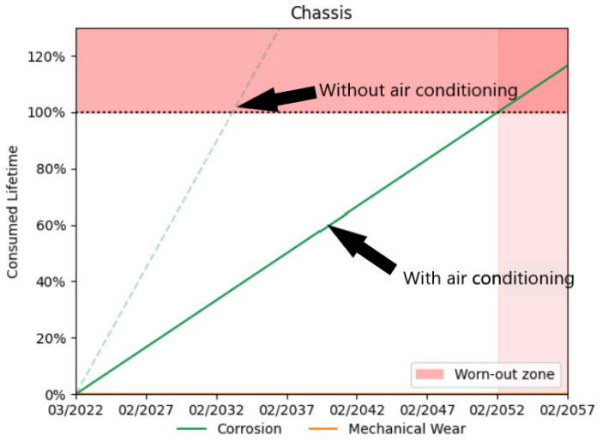

Fig. 2 Air conditioning impact

C. What-if analysis: alternate maintenance scenarios

Concerning maintenance, what-if scenarios allow to optimize the frequency of maintenance plans that can be adapted to each customer.

Most often, maintenance plans are extracted from maintenance guides, based on fixed periodicity recommendations [9]. The described approach enables the generation of adequate matches to plan maintenance operations, and advise our customers about the risk taken by withholding manufacturer maintenance. This is possible by simulating the life operations of the customer asset, and then performing maintenance on a subcomponent only when it reaches its end of life by comparing two scenarios : the standard maintenance plan and the custom one depending on ageing. We have compared these two scenarios in Fig. 3, where the effects of maintenance actions are reflected on the curve by sudden drops in lifetime consumption.

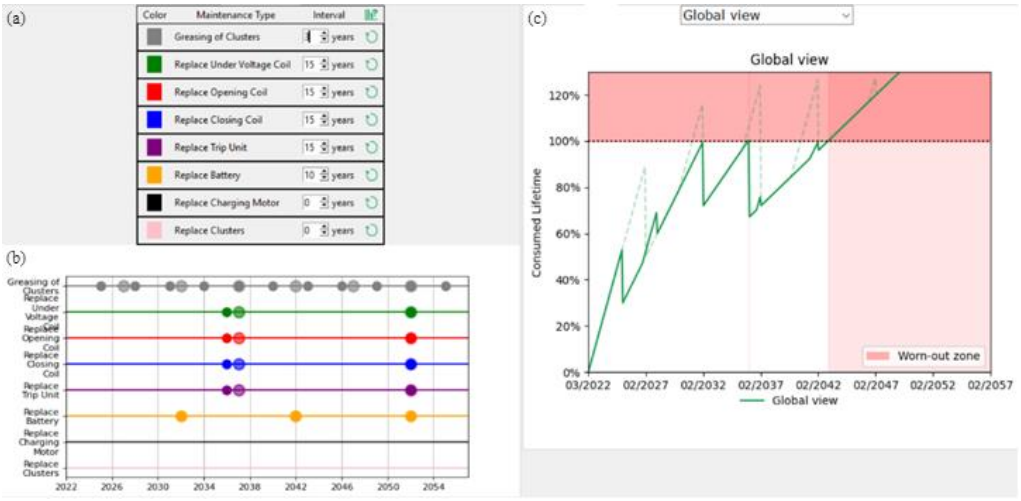

Fig. 3 Maintenance impact

Fig. 3 (a) shows the maintenance periodicity selector used to configure the maintenance scenario. (b) illustrates how the selected scenario differs from a reference scenario and provides maintenance periodicity fine-tuning capabilities. Finally, (c) provides a comparison of the selected scenario with the reference one in terms of ageing.

D.Field deployment

The previous section on what-if scenarios described how simulation helps the validation step, in terms of both ageing simulation and maintenance scenario.

These simulation processes allow to transform traditional maintenance practices into a condition-based maintenance approach. Indeed, maintenance manufacturer periodicity recommendation described in maintenance guides are useful to provide an overview of the equipment lifetime. However, depending on type of segments (marine, mining-minerals-metals, oil and gas, healthcare, food & beverage), constraints are not the same. Two examples are described below:

- Marine customers have the specific constraint to make maintenance action when boats return to dock. Being able to estimate the ageing of the assets and consequence of maintenance actions facilitates the anticipation of required actions, ensuring confidence in the asset behavior waiting for the next maintenance while cruising.
- Regardless of the type of segments, the number of equipment is not the same from one customer to another. When the number of equipment is high, this kind of simulation helps to manage the fleet maintenance management by anticipating consequence of maintenance rescheduling for part of the equipment.

Therefore, combining manufacturer periodicity recommendation and simulation process allows to adapt the maintenance actions to each customer by adding more flexibility, more anticipation of end-of-life, and making timely maintenance. This condition-based maintenance approach makes operations more efficient while making business more resilient and sustainable [12].

With Schneider Electric, this approach is used by customers from different segments and for different kinds of assets, on low voltage or medium voltage domains. In the coming two years, we expect to grow by 200% the number of customers asking to transform traditional maintenance practices into condition-based maintenance ones. By increasing the number of field experiments, we will continuously challenge and improve the current models of ageing estimation, leveraging the benefit to link simulation and validation by closing the loop from customer, fostering its data to improve the models.

III. SHORT-TERM

Section II showed how what-if simulations can help decide and validate some maintenance actions to reduce long-term failures, or increase expected lifetime. This section focuses on short term analysis to detect unexpected failures. We first remind the proposed approach using machine learning model, then describe the incremental learning strategies, and finally illustrate with experimental results.

A. Proposed approach

Virtual sensors are Machine Learning-based techniques used for short-term anomaly detection [1]. Virtual sensors predict various industrial data based on usage and environmental conditions. Such idea of "virtual sensor", also known as "soft sensor", is not new and has been described in many ways in the literature and implemented in industrial products (see [2] for a review). A virtual sensor can predict a category (classification) or a quantity (regression). Besides their use for better process control, virtual sensors are used to tackle many other problems, such as back-up of a real sensor, what-if analysis, sensor validation, and fault detection and diagnosis.

The latter consists typically in monitoring a statistically significant deviation between the actual monitored data and the learnt reference provided by the virtual sensor. A usual way to perform this step is to model the prediction error (so-called residuals) and use this model to detect a statistical outlier [5]. Fig. 4 illustrates how such Virtual Sensor Fault Detection (VSFD) technologies can be assembled to create an adaptive temperature monitoring solution.

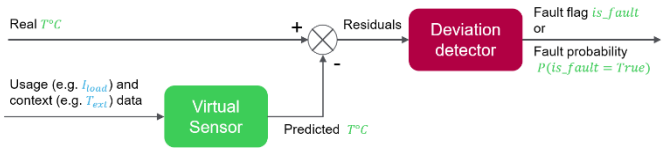

Fig. 4 Virtual Sensor Fault Detection Principle

B. Incremental learning framework

1) Context

Virtual sensor models are learnt on historical data capturing normal process operation. Such historical data spans over a limited period of time and therefore may describe a limited set of process operations. New data may contain unseen normal samples representing other or new aspects of the process. This notion known as *concept drift* is affecting the predictive accuracy of models over time. *Adaptive* Soft (Virtual) Sensors approaches [13] propose to update models using the new samples, or fully retrain models with the augmented dataset. Advanced sample selection and weighting techniques, as well as ensemble methods, are typically used to cope with challenging drifts [7]. Finally, recent uses of deep learning models to perform such tasks highlighted a specific issue known as the stability-plasticity dilemma, leading in extreme cases to *catastrophic forgetting* [10].

More general frameworks known as *Online*, *Sequential*, *Incremental*, *Lifelong*, or *Continual* learning [3] have recently emerged to describe applications where multiple aspects of the problem evolve with time. While the vast majority of work focuses on deep learning models for image classification, some papers mention regression tasks [6]. Adaptive virtual sensors can be seen as a particular case of incremental learning with a single task (regression), incremental on the data-domain.

Based on this review, we can delineate three general concerns in incremental learning regarding updates:

- *When* to update: model updates can be set to run periodically (e.g. weekly), or triggered whenever novelty is detected in the samples by a novelty detection model [14]. An alternate approach is to store novel samples in a buffer and update the model when maximum capacity is reached [6]
- *What* samples to integrate: too large new sample batches may be sub-sampled; under-represented class labels may be boosted with sample generation techniques; etc.
- *How* to inject the new samples: special update steps can include sample weighting for example

Most of the literature on incremental learning considers all samples to be safe for reinjection and is concerned with how to inject them. Some applications of adaptative virtual sensors are concerned with *what* data to reinject ; they tackle the issue using specific models and architectures [14]-[15]. These studies focus on classification models; besides they do not try to evaluate and compare different update strategies – but implement one and assess its performance.

There is no literature to date, to the best of our knowledge, comparing incremental learning strategies for regression virtual sensors for fault detection tasks (III.A). Process faults arising in operational data are an extreme case of outliers or concept drift: the statistical properties of the data during fault periods become different from the normal one used in the training set. However, as opposed to the usual concept drift handling approaches, the updated model should be protected from deviations so that it can still detect such faults in the future. In other words, new samples to integrate in the model should be carefully curated to eliminate abnormal samples and preserve its aim at modeling normal behaviors.

2) General Framework

An initial model M_1 is learnt on an initial dataset D_1. D_1 is considered safe: it only contains normal (non-fault) data. The following steps are then performed in sequence:

1) Use model M_1 to predict the target variable and detect faults during next period (dataset D_2)
2) Based on step 1 outcome, filter dataset D_2 to keep only "safe" samples D_2'
3) Update model M_1 with D_2' to create model M_2
4) Iterate: repeat step 1-3 with new model M_2 and next period D_3

Fig. 5 below illustrates this procedure. The curation step (2) is indicated with a star (*).

Note that this procedure describes online learning-style update steps. It can be easily adapted to include full model retraining instead of model updates. Associated alternate Step 3 becomes:

3') Append D_2' to previous dataset D_1. Train new model M_2 using the merged dataset.

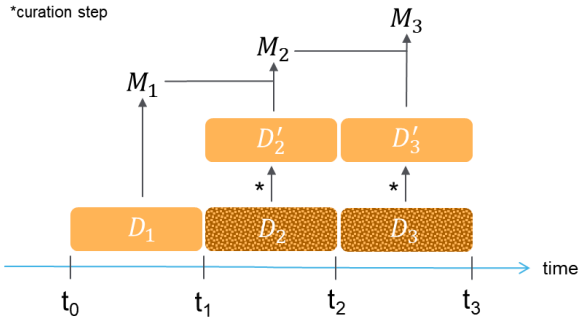

Fig. 5 Incremental learning framework

3) Considered implementation

Within the framework described in III.B.2), we consider the particular implementation below. An initial training period of 4 weeks is considered for D_1. Incremental learning is done on a weekly calendar basis. The curation step (2) is composed of two sub-steps:

- *Automated faults filtering*: even when the model predicts a fault, some samples are automatically re-tagged as normal.
  - *Level 1*: previously unseen (out-of-training range) data is re-tagged as normal,
  - *Level 2*: in addition to level 1, transient faults and faults tagged as unrealistic by an expert rule are re-tagged as normal. Such an expert rule includes physics-driven concerns, for example in some contexts an "under-heating" fault is not likely to be an actual fault but rather a model prediction error.

- *Faults validation*: faults are then presented to a human operator (expert). This operator is in charge of providing the final label: it can either "confirm the fault" or "discard the fault". Discarded faults are re-tagged as normal.

Finally, models are fully relearnt (3'.) instead of being updated (3.), so as to eliminate suboptimality issues related to partial updates. The size of the datasets is small and the considered model simple, thus full retraining is not prohibitive; and neither security nor storage are issues here. The overall process is described in Fig. 6.

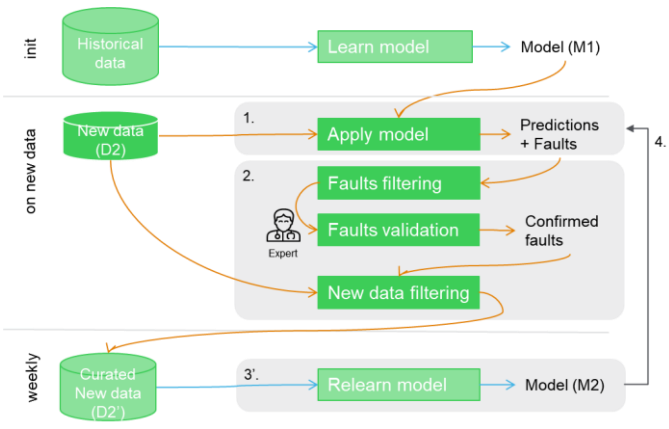

Fig. 6 Virtual Sensor Fault Detection orchestration

C. Validation using simulation

Validating the adaptive VSFD orchestration presented in III.B.3) is challenging, as only a little real incident data is available from the field. To overcome these difficulties, we propose to simulate the entire procedure on realistic datasets. This allows us both to validate correct behavior (overall feasibility) and to choose between algorithmic configurations.

1) Annotated datasets

Datasets injected in the simulator contain actual measurements (a multivariate timeseries) recorded from the field during several months of process operation. Each dataset is annotated: it is associated with zero, one or several *fault periods*. A fault period consists of a start and end date, and corresponds to a real incident observed on the field. Fault periods are usually defined as broader than the actual fault, because it is assumed that unobserved fault samples may already be present before the fault has been detected by a human operator. Also, logbooks from the field may be imprecise and contain vague fault period definitions.

2) Algorithm configurations

We consider eight alternate virtual sensor configurations related to the size of sliding window used: 1h, 2h, 3h, 6h, 9h, 10h, 12h, and 24h. For a given virtual sensor configuration, we consider 4 alternate fault detection threshold multiplicator (see [5]): 2, 2.5, 3, and 3.5.

3) Update strategies

Although most steps described in previous section III.B.3) present no particular technical difficulties, a challenge comes with simulating the human expert contribution happening in step 2 (fault validation). Besides, the *Level 2* automated faults filtering step may mistakenly re-tag actual fault samples as normal ones. We define below three strategies to simulate the fault filtering and validation process: *conservative*, *realistic*, and *oracle*.

Conservative strategy: the conservative strategy represents a low-risk scenario. Only *Level 1* fault filtering is active, and all other faults keep their label. *Level 2* is inactive and the expert always confirms faults. In other words, only samples not tagged as fault by the VSFD and samples out-of-range of the training space are considered normal and used to relearn the model. In this scenario, a minimal volume of new data is reinjected for learning every week.

Realistic strategy: this strategy is quite similar to the conservative one. This time, both *Level 1* and *Level 2* fault filtering are active, but the expert cannot discard faults. In other words, in addition to the conservative strategy, samples representing non-physically valid or transient faults are reinjected too. In this scenario, a medium volume of data is reinjected for learning every week.

Oracle strategy: in the oracle strategy, both *Level 1* and *Level 2* fault filtering are active, and the expert is omniscient and knows exactly when actual faults occur. In other words, in addition to the realistic strategy, samples detected as fault and not re-tagged by the filtering are automatically re-tagged as normal by the expert if they fall outside the actual fault period. In this scenario, a maximal volume of new data is reinjected for learning every week.

The conservative and oracle strategies have opposite behaviors and serve as representative bounds of the reality. In the conservative and realistic strategies, the expert is not confident on its abilities to recognize faults, while in the oracle scenario, it never makes mistakes. In the real world, experts are somewhere between realistic and oracle. They can adapt their behavior to particular customers and assets, as well as use their expertise and analysis skills on the measurement timeseries to discard false alarms. Simulating such an adaptative strategy for experts is out of scope of this study.

4) Assessment metrics

We introduce the following performance metrics to evaluate how well a given simulation run has succeeded in terms of fault detection.

Each sample is tagged as *False Positive* (FP), *True Positive* (TP), *False Negative* (FN) or *True Negative* (TN) depending on whether its label after the *Automated faults filtering* step matches the ground truth label (True/False) and if it is fault (Positive) or normal (Negative).

The *fault detection indicator* FD is defined as a boolean/dummy variable, equal to 1 when at least one fault was detected within the fault period, and to 0 otherwise.

The *fault periods coverage* FPC (a.k.a. *sensitivity* or *recall*) is defined as the ratio between the number of samples tagged as fault in the fault periods and the total number of samples in the fault periods.

The *false alarm ratio* FAR (a.k.a. *false positive rate*) is defined as the ratio between the number of samples tagged as fault outside of the fault periods and the total number of samples outside the fault period.

$$FPC = \frac{TP}{TP + FN} \qquad FAR = \frac{FP}{FP + TN}$$

5) Experimental results

We use data from three customer assets: two low-voltage panels and one medium-voltage panel. Inside a panel, a data set may be representative of different sub-assemblies: cable, busbar or withdrawable circuit breaker connections. Datasets span several months and contain either zero or one annotated fault period, that may be approximate (see III.C.1). Table I describes the five datasets used in this study.

We run the procedure described in (III.B.3): for each of the five cases, eight algorithm configurations, four fault detection threshold multiplicators, and three update strategies; giving in total 480 alternatives to evaluate. The *FPC* and *FAR* metrics are computed on each alternative. They are then averaged across all datasets, so that each model configuration is associated with a single pair of metrics.

Fig. 7 presents the results obtained in this experiment. For each of the considered update strategy (subplots (a), (b), (c)), the *FPC* (x-axis) and *FAR* (y-axis) are displayed for all alternatives algorithm configurations. Alternatives are grouped by sliding window size (colored line), where each group

TABLE I
DESCRIPTION OF USE CASES (15 MINUTES DATA SAMPLING RATE)

Case	Customer	Asset	Sub-assembly	Fault period	Dataset duration	Observations
1.1	1	LV Panel	N° 1	01/03/2019 – 15/04/2019	04/06/2018 – 08/03/2021	96,799
1.2	1	LV Panel	N° 2	No fault	04/06/2018 – 08/03/2021	96,799
2.1	2	LV Panel	N° 1	No fault	10/07/2019 – 11/03/2020	23,497
2.2	2	LV Panel	N° 2	01/10/2019 – 31/10/2019	10/07/2019 – 11/03/2020	23,497
3.1	3	MV Panel	N° 1	01/09/2020 – 30/09/2020	14/11/2019 – 24/03/2021	47,620

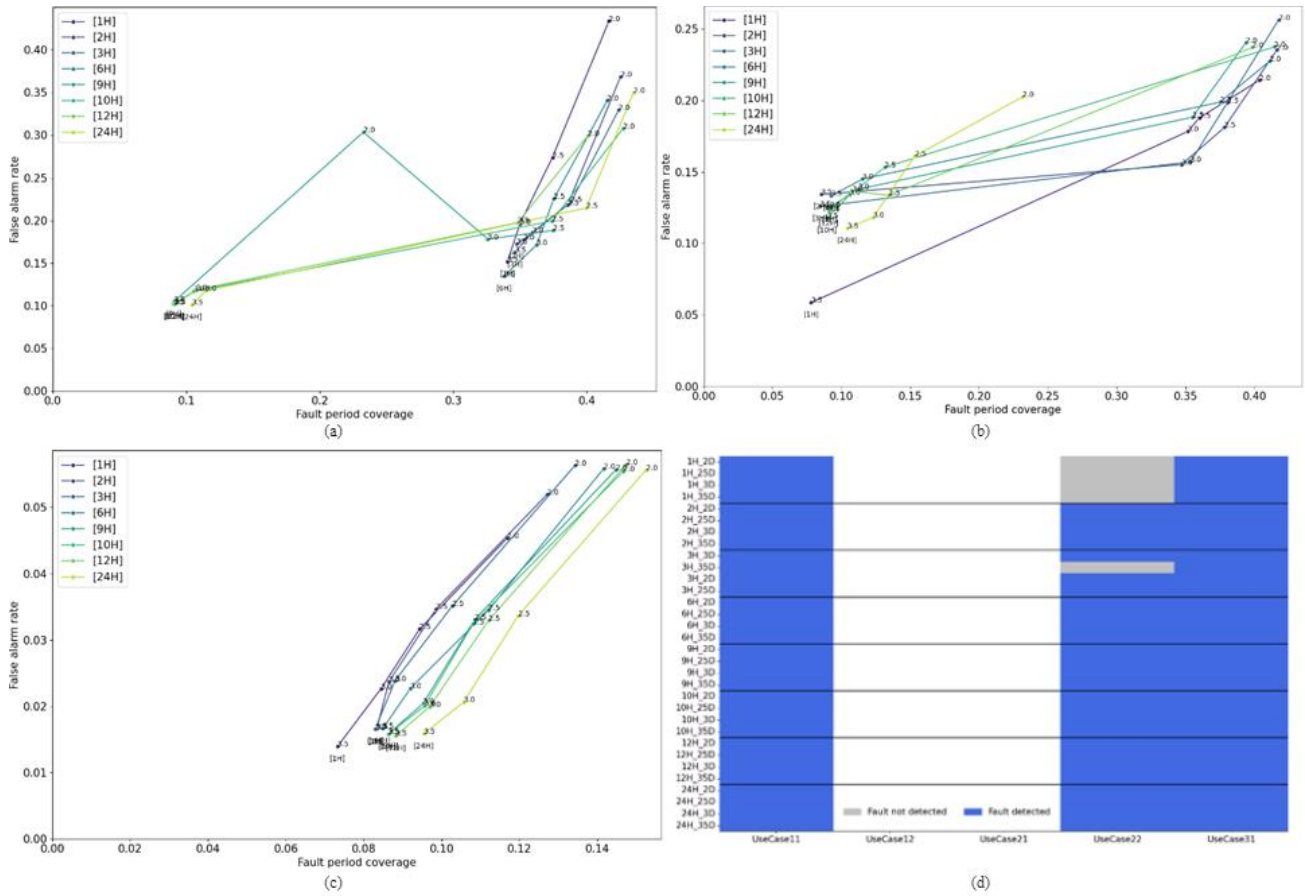

Fig. 7 Results for the Conservative (a), Realistic (b) and Oracle (c) strategies.
Fault detection heatmap (d)

contains results for the 4 fault detection multiplicator values. In addition, a fault detection heatmap for FD is computed for each strategy; however since all lead to identical results a single one is presented (d).

From the fault detection (*FD*) heatmap, we can first discard configuration [1H] (models using a sliding window size of 1h): whatever the fault detection multiplicator, none was able to detect the fault in use-case 2.2. This is also the case for configuration [3H] with multiplicator 3.5.

From the scatter plots, we see several trends. First, the amount of false alarms (*FAR*, y-axis) decreases as the strategy moves from Conservative (10-44%) to Realistic (5-25%) and finally Oracle (1-6%). This tends to confirm the role of a good expert feedback loop in the quality of incremental learning.

Concerning the fault prediction coverage (FPC, x-axis),

- In the *Oracle* scenario (c), all models have similar values (~10%). This consensus seems to indicate that the *actual* fault period is much smaller than the one declared in the datasets. In addition, we see that large sliding windows have better coverage: they are closer to 10%, while small ones are close to 5%. For the rest of analysis below, we use 10% as the optimal coverage to reach and consider higher values as "erroneously better".
- With the *Realistic* strategy (b), we see two groups of models with similar values (left: ~10%, same as in Oracle, or right: ~40%). The left group ("correct") corresponds to high fault detection multiplicators values, while the right group corresponds to low ones. In this scenario, the results

therefore tend to indicate that all models can be moved to the left group simply by increasing the fault detection multiplicator.

- With the *Conservative* strategy (a), the same two groups are visible, with the same values. However this time, models with small sliding window sizes (6h or less) are not able to move to the left group, even when the fault detection multiplicator is increased. It seems to indicate that to tackle this worst-case scenario, large sliding windows are preferable. We also note on this chart the presence of an outlier point.

Overall these results highlight a better stability of models using large sliding windows (>6h) and a large fault detection threshold multiplicator (3.5), even when the worst-case Conservative strategy is used. Adding *Level 2* filtering (III.B.3) brings a lot of value in terms of false alarms rate reduction and models stability. Finally, since these results were obtained on only five datasets, two of which without fault periods, it might be relevant to pursue larger-scale analysis to see if these trends are confirmed.

IV. Conclusion

Condition monitoring of critical assets requires both a short-term and a long-term perspective in order to both detect sudden anomalies and slow degradation. At Schneider Electric we combine Machine Learning methods for the short-term, and First principle statistical methods for the long-term. Validating such approaches is challenging. Indeed most of the real world collected data captures normal behavior and does not span for a long enough period of time to capture the asset degradation.

In this paper we propose to use simulation to overcome these difficulties:

- Long-term degradation simulation requires to simulate the environment and usage over decades of product life. Simulating the environmental conditions (Temperature, Humidity…) is performed using both timeseries modeling techniques (e.g. ARIMA) and modifications of real timeseries captured from sensors. Simulating usage is performed using modifications of real load profiles, representative of the diversity of use cases. Finally, several maintenance scenarios can be simulated in order to benchmark their efficiency: current field practices, optimized maintenance, etc.
- For Short-Term Machine Learning-based approaches, we can use actual field data in the simulator as a few months/years are sufficient to see practical convergence and accurate fault detection. However the challenge is associated with simulating model incremental learning over time, especially when a human operator is supposed to confirm/discard faults. We propose to bound the space of exploration with a pessimistic and an optimistic human behavior simulator, as well as a "probable" median scenario.

References

[1] Chevalier, M., S. Marié, B. Boguslawski, M. Cercueil, F. Chupot, A. Vignon, and W. Youssef. 2021. "Combining First Principles and Machine Learning for optimal maintenance of electrical assets". In *CIGI QUALITA 2021*. May 2021, Grenoble, France.

[2] Curreri F., G. Fiumara, and M. Xibilia. 2020. "Input Selection Methods for Soft Sensor Design: A Survey". *Future Internet* vol. 12 , no. 6: 97. https://doi.org/10.3390/fi12060097.

[3] Delange, M., R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, ... and T. Tuytelaars. 2021. "A continual learning survey: Defying forgetting in classification tasks". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. doi: 10.1109/TPAMI.2021.3057446

[4] Crawley D. B. and K. L. Lawrie. 2019. "Should We Be Using Just 'Typical' Weather Data in Building Performance Simulation?". In *16th IBPSA Conference*, 2019. https://climate.onebuilding.org

[5] Gao, T., B. Boguslawski, S. Marié, P. Béguery, S. Thebault, and S. Lecoeuche. 2019. "Data mining and data-driven modelling for Air Handling Unit fault detection". In E3S Web Conf., Volume 111, *CLIMA 2019 Congress*. Bucharest, Romania.

[6] He, Y. and B. Sick. 2021. "CLeaR: An adaptive continual learning framework for regression tasks". *AI Perspectives* vol. 3(1), pp. 1-16.

[7] Kadlec, P., R. Grbić and B. Gabrys. 2011. "Review of adaptation mechanisms for data-driven soft sensors". *Computers & chemical engineering* vol.35(1), pp. 1-24.

[8] Kegel. L, M. Hahmann, and W. Lehner. 2017. "Generating What-If Scenarios for Time Series Data". In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management* (SSDBM '17). Association for Computing Machinery, New York, NY, USA, Article 3, pp. 1–12.

[9] Masterpact MTZ Maintenance guide, 2015 https://www.se.com/ww/en/download/document/0613IB1202/

[10] Parisi, G. I., R. Kemker, J. L. Part, C. Kanan and S. Wermter. 2019. "Continual lifelong learning with neural networks: A review". *Neural Networks* vol. 113, pp. 54-71. ISSN 0893-6080

[11] Rizzi S. (2009) What-If Analysis. In: LIU L., ÖZSU M.T. (eds) *Encyclopedia of Database Systems*. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-39940-9_466

[12] Schneider Electric, 2021, https://www.se.com/ww/en/work/services/service-plan/ecostruxure-service-plan.jsp

[13] Souza, F. A., R. Araújo and J. Mendes. 2016. "Review of soft sensor methods for regression applications". *Chemometrics and Intelligent Laboratory Systems* vol. 152, pp. 69-79.

[14] Carino, J. A., et al. 2018. "Fault Detection and Identification Methodology Under an Incremental Learning Framework Applied to Industrial Machinery," in *IEEE Access*, vol. 6, pp. 49755-49766.

[15] Yu, Y., Peng, M., Wang, H., Ma, Z., Cheng, S., & Renyi, X. 2021. "A continuous learning monitoring strategy for multi-condition of nuclear power plant". *Annals of Nuclear Energy*, 164, 108544.