Filtering Misleading Repair Log Labels to Improve Predictive Maintenance Models

Pablo del Moral[1], Sławomir Nowaczyk[1], and Sepideh Pashami[1,2]

[1] *Center for Applied Intelligent Systems Research CAISR, Halmstad University*
pablo.del_moral@hh.se
slawomir.nowaczyk@hh.se
sepideh.pashami@hh.se

[2] *RISE Research Institutes of Sweden*
sepideh.pashami@ri.se

ABSTRACT

One of the main challenges for predictive maintenance in real applications is the quality of the data, especially the labels. In this paper, we propose a methodology to filter out the misleading labels that harm the performance of Machine Learning models. Ideally, predictive maintenance would be based on the information of when a fault has occurred in a machine and what specific type of fault it was. Then, we could train machine learning models to identify the symptoms of such fault before it leads to a breakdown. However, in many industrial applications, this information is not available. Instead, we approximate it using a log of component replacements, usually coming from the sales or maintenance departments. The repair history provides reliable labels for fault prediction models only if the replaced component was indeed faulty, with symptoms captured by collected data, and it was going to lead to a breakdown.

However, very often, at least for complex equipment, this assumption does not hold. Models trained using unreliable labels will then, necessarily, fail. We demonstrate that filtering misleading labels leads to improved results. Our central claim is that the same fault, happening several times, should have similar symptoms in the data; thus, we can train a model to predict them. On the contrary, replacements of the same component that do not exhibit similar symptoms will be confusing and harm the ML models. Therefore, we aim to filter the maintenance operations, keeping only those that can be used to predict each other. Suppose we can train a successful model using the data before a component replacement to predict another component replacement. In that case, those maintenance operations must be motivated by the same, or a very similar, type of fault.

We test this approach on a real scenario using data from a fleet of sterilizers deployed in hospitals. The data includes sensor readings from the machines describing their operations and the service logs indicating the replacement of components when the manufacturing company performs the service. Since sterilizers are complex machines consisting of many components and systems interacting with each other, there is the possibility of faults happening simultaneously.

1. INTRODUCTION

In this paper, we are going to deal with the common industrial problem of learning predictive maintenance models using labels from repair logs. This work has been inspired by the collaboration with our industrial partner Getinge AB, a company producing sterilizers to be used at hospitals for medical equipment. The machine sterilizes its load by using phases of low pressure to eliminate air and humidity combined with phases of high temperature that kill micro-organisms. A sterilizer is critical for the hospital's operation: without properly sterilized material, most daily activities can not be executed. Getinge also provides service and maintenance. Unexpected failures often lead to long downtimes: a technician needs to be sent to the machine, the problem diagnosed, the right parts ordered and installed, and the machine needs to be tested.

The ideal scenario to train a predictive model is to monitor a machine while undergoing a fault, record the data describing its operation in the meantime, and find patterns that describe the effect or symptoms of this fault. Later, these patterns can be used to identify similar faults in the future so that maintenance actions can be performed before the issue leads to an unexpected breakdown. This avoids consequential or collateral costs associated with a breakdown without the increase in maintenance costs associated with preventive maintenance

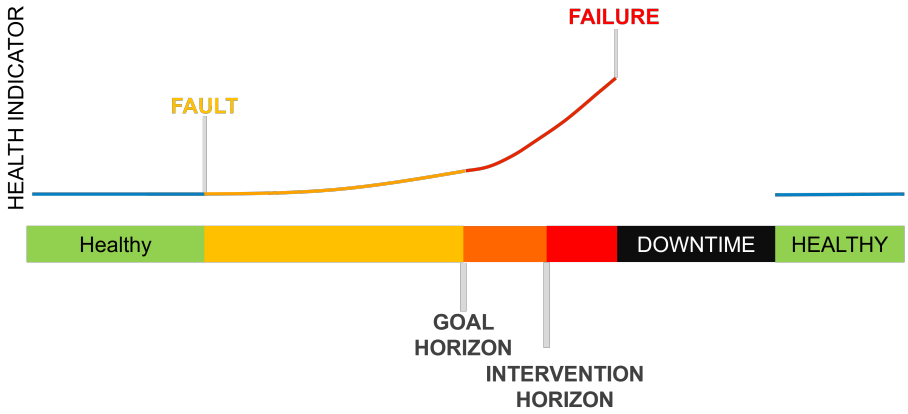

Figure 1. An idealized overview of the setup and the data needed to train a predictive maintenance model for estimating Remaining Useful Life.

schemes.

This ideal scenario is possible to realize during the design of the machine but relatively difficult during real-world operation. Analyzing different faults under controlled conditions can be done in a lab setting while designing and deciding what data to record to monitor the evolution of the fault properly. However, in our application scenario and many similar ones, this ideal scenario is not feasible. First, tests under controlled conditions do not necessarily reflect the real operation of the machines in the field. External conditions such as water temperature, humidity, load, and usage considerably affect the functioning of the machine. Reconstructing all of them in the lab is a daunting task, not viable economically. Even more importantly, our industrial partners need to develop predictive maintenance models on already built and deployed machines. Like many other machines, sterilizers are expected to work for many years, often decades; and it is precisely those older machines that have the room for the greatest gains from a new, ML-based, successful predictive maintenance solution.

In other words, we would like to learn predictive maintenance models using data from the machines that are already deployed in the field. These machines have been collecting sensor data for several years, mainly for purposes of control and security. We can use this data to find patterns to predict future failures – however, one has to keep in mind that the data is far from ideal since it has generally not been collected for the purpose of monitoring the health of the machines.

Another aspect to consider is that we do not have direct, reliable information about the faults that happened in the machines. However, we need at least some approximation of a history of past faults to be able to label the sensor data and build predictive models. In reality, the best information that we can typically get is the maintenance logs or repair histories, i.e., a register about when maintenance was done and which components were installed in each machine. This way of labeling the data is very unreliable for various reasons that we will present in detail in the next section; if used naively, it necessarily leads to inaccurate models.

In this paper, we present a methodology to filter these repair logs, selecting only the labels that can be used to provide reliable predictions. The key concept is based on the following observation. If the same *predictable* fault happens in several machines, similar patterns will be found in the sensor data. These faults will develop into failures; then maintenance will be carried out, and components will be replaced; we will obtain information about those component replacements (only, not of the faults directly) through the maintenance logs. Our underlying assumption is that the inverse should also hold. Suppose we can identify patterns in the data before a component replacement that are useful to predict other component replacements. In that case, those component replacements must be related to a similar type of fault. This approach, on its own, would be very prone to overfitting; therefore, we add a second step to our method, where we refine our models by adding the false alarms created in the first step.

The rest of the paper is structured as follows: in Section 2, we motivate our research, specifically focusing on why the service logs are often unreliable; in Section 3, we present the different steps of our approach; in Section 4, we summarize our experiments and discuss the results; in Section 5, we compare our work against state of the art and provide the literature review; and in Section 6, we present our conclusions and future work.

2. Problem Formulation

This section will describe some of the problems typically encountered while labelling predictive maintenance data using information coming from service logs.

Before we focus on the labels, though, it is important to note that the data recorded in the machines is generally not designed to describe the health of the machine and its components accurately. Therefore, it is expected that many faults will show no symptoms in the data or will show confusing symptoms. Our task is to find as many patterns as possible in this data that could be linked to faults that later lead to failures.

In Figure 1, we have an example of the idealized training case to create models that predict faults. The machine is in a healthy state until a fault happens. We are recording a signal (one or multi-variate) that perfectly describes the machine's state of health. Once the fault is introduced, this health begins to deteriorate, and the progression is reflected in the selected health indicator. At some time, it reaches the failure point, and the machine stops working. After the failure, a repair is needed, which means that the machine will be out of operation until it is repaired. Then, the machine goes back to operation at full health.

Planning and executing a maintenance operation requires time. We define this time as the "intervention horizon": the last moment before a failure when it is possible to intervene on the machine and avoid it. The goal horizon adds a safety margin to schedule the maintenance operation (with perfect models, the goal horizon and the intervention horizon would be the same). We are recording data to obtain a health indicator; we can label this data and train a classifier with it based on the goal horizon. The data before the goal horizon would be "SAFE," and the data after the goal horizon would be "ALARM." In the future, when the model predicts an "ALARM," we can schedule a maintenance operation in the most convenient way and avert the failure.

However, in real-life applications, when we must rely on using maintenance records, this ideal case does not happen. The only information we have is when a component was replaced, in a given machine. This presents a number of ambiguities:

- There are uncertainties in the dates. Even assuming that the dates are entered correctly (which is not always the case due to human error or accounting policies), we know when a component was replaced in the machine. However, we do not know how long it took from the failure to the component replacement.
- In fact, we do not know if the replaced component had failed or not. The replacement of a component can be due to a failure, but it can also be due to a preventive maintenance operation or a subjective decision by the technician.
- If there was a fault happening in the machine, we do not know if the replacement of the component solved it. In other words, we can not be sure that the diagnosis by the technician doing maintenance was correct.
- If many components were replaced, we do not know which one was responsible for the fault or if different faults were happening simultaneously. Usually, multiple symptom patterns can be identified in the data – and matching them to replaced components is error-prone.
- We do not have the certainty that all the maintenance operations are recorded in the service logs. Some of the maintenance operations can be carried out by the staff operating the machines. In addition, hospitals can buy service from different companies.
- In the case where the failure actually happened and the responsible component was replaced, we do not have information about when the fault started.

These uncertainties will lead to wrong labeling of the sensor data recorded. The effect of such wrong labeling is particularly harmful because we are not just labeling one data point but a full sequence of data, from the moment the fault starts, until the failure happens.

To sum up, using the maintenance records, the only certainties we can obtain are the intervention and goal horizons. We can use the goal horizon to label the recorded data by the machine as "ALARM" until the component is replaced. We can label an arbitrary amount of the data before the goal horizon as "SAFE."

3. Literature Review

Predictive maintenance is a hot topic in the research and industrial communities. According to (Thomas & Weiss, 2020), maintenance strategies based on corrective resulted in more than 3 times more downtime and 16 times more defects than more advanced maintenance strategies.

A quick overview of recent surveys (Carvalho et al., 2019), (Lei et al., 2020), shows how most data-driven methods for predictive maintenance use supervised methods, i.e., reliable information about the historical faults are needed to train models for predictive maintenance.

There are different approaches to obtaining accurate labels. One solution is to use simulated data, where the operation of a machine or system is simulated, and faults are introduced at known times. Examples of these datasets are the "Turbofan Engine Degradation Simulation Data Set" (Saxena & .K, 2008) or the "Tennessee Eastman Process" (Ricker, 1996).

Another option is to run tests done in a laboratory (Yang, Stronach, MacConnell, & Penman, 2002). In the field of fault prediction for bearing machinery, this approach is very popular: a setup is built, data is recorded, and faults are introduced. Again, the moment when the fault was introduced is clearly determined, and the evolution of the fault is carefully monitored.

The problem of using repair logs as a reliable source of infor-

mation has also been researched. In (Seale et al., 2019), the authors present an approach to complete the information of the repair logs using Natural Language Processing approaches to determine which component was the recipient of a particular maintenance operation.

In (Prytz, Nowaczyk, Rögnvaldsson, & Byttner, 2015), the authors describe some of the uncertainties coming from using the information from the repair logs as a source of labels for the data recorded in trucks. Although not focusing directly on this problem, they study how uncertainties in the dates can have a significant role in setting parameters such as the prediction horizon.

Finally, in (Sipos, Fradkin, Moerchen, & Wang, 2014), the authors use log data from medical equipment to predict future failures. They discuss the problems of relying on the information coming from the repair logs and build their approach to solve this problem. The main difference between their problem definition and ours is that they can assume that the absence of repair logs for a given time means that the machine is healthy, while we can not make this assumption in our practical case.

From the technical point of view of machine learning, our scenario is related to the task of learning in the presence of noisy labels. According to the taxonomies presented in (Frénay & Verleysen, 2013), we can categorize our approach as "model predictions-based filtering" since we use the performance of a classifier as a tool to filter the label noise. As an example, in (Nguyen et al., 2019), the authors use the output of a neural network during the training to detect the noisy labels and filter out the corresponding instances. The main difference between our approach and the state-of-te-art stems from the nature of the noise in our data: the uncertainty resides in the repair logs, which are used not just to label a single instance, but a complete sequence of instances.

4. METHOD

4.1. Data

For this study, we work with the data coming from 67 machines situated in several different countries. The data coming from the machines contain sensor data measuring magnitudes such as pressure or temperature inside the chamber of the sterilizer during its processes. Although all the sterilizers are part of the same product line, there exist different models with, for example, different sizes of the sterilization chamber or different versions of particular components.

We will use two years of recorded data produced by the machines. Not all the machines have produced data for the whole period, and most of them have been deployed on the field much longer than these two years. For each process of the machine (usually called cycle), the raw data contains sensor reading for the pressure and the temperature in different parts of the chamber. Working with the experts, we have extracted 86 features that characterize each cycle. Typically, a machine runs 5-10 cycles per day. Per regulation, one of these cycles has to be run on an empty load to validate with certain biomarkers that the machine is still achieving sufficient sterilization.

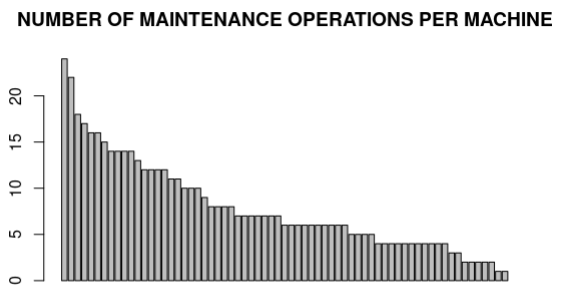

Figure 2. Number of maintenance operations per machine.

For those two years of data, a total of 275 maintenance events with component replacements have happened. In Figure 2, we can see the distribution of the number of maintenance events with component replacement per machine. Some machines have very few component replacements logged in the service logs, while others have a larger number.

4.1.1. Data Preprocessing

Every machine has a slightly different configuration and can be used in different settings. For example, different machines can have different sizes of the chamber, which obviously affects how the extracted features from the data look like. But even for the same machine, conditions can change over its lifetime, for example, by upgrading one of the components or due to changes in the usage patterns.

After discussion with the experts, two factors need to be considered. First, the data is subject to small random variations based on factors like the room's water temperature and humidity, among others. Second, to normalize the data across machines, we should focus on the rate of change of the features, not on their absolute values.

With this in mind, we first calculate a moving average of all signals to reduce the effect of small random variations in the data. For every cycle, the value for each feature is the average over the previous 100 cycles. Then, we take the difference between the value of the moving average of the current cycle and the cycle 100 cycles before. Thus, the final value for the features of a given cycle is calculated by considering the previous 200 cycles.

4.2. Filtering of Component Replacements

Based on the discussion in Section 2, we label the data 75 cycles before a component replacement as "ALARM" and the data between 150 and 75 cycles before the component replacement as "SAFE." This corresponds approximately to 15 and 30 days before the component replacement. This is in line with the time it would take to schedule and perform maintenance on the machine.

Our assumption is that if we can train a classifier on the data before a component replacement, and use that classifier to accurately predict another component replacement, then those two component replacements are most likely related to the same fault.

Algorithm 1 Fitness function.

N set of events
$D_i = [(\mathbf{x^1}, y^1)...(\mathbf{x^m}, y^m)]$ dataset for event i
R vector of results
for $i = 1$ to $length(N)$ **do**
    $D_{train} = \bigcup_{j \neq i} D_j$
    $m = \text{trainclassifier}(D)$
    $\mathbf{pred} = \text{predictprobability}(D_i)$
    $R_i = \text{measureAUC}(\mathbf{pred}, \mathbf{y_i})$
end for
F: fitness value
for $r = 1$ to $length(N)$ **do**
    if $R_i >$ threshold **then**
        $F = F + 1$
    else
        $F = F - 1/3$
    end if
end for

With a number N of component replacement events available, our goal is to select the largest subgroup of events that can be used to train classifiers that can predict each other. Each event i is associated with a training set $D_i = [(\mathbf{x^1}, y^1)...(\mathbf{x^m}, y^m)]$, where $\mathbf{x^i}$ is the feature vector, and y is the associated label ("SAFE" or "ALARM"). The number of possible subgroups increases dramatically as we consider more and more component replacement events. To perform this search, we use a genetic algorithm. In Algorithm 1, we present the fitness function used. The goal is to select as many events as possible such that they can all be used to predict each other, while discarding as many events as possible that can not be predicted.

To implement the genetic algorithm, we use the *ga* function from the **GA** package in R. The population at each generation is 50, the probability of cross-over is 0.8, the probability of mutation is 0.1, and the elitism is set at 0.1. We choose to stop the search after 10 iterations without improvements in the best solution.

4.3. Experimental Setup and Evaluation.

We split the two years of data into four periods of half a year each. For each period, we train our models with all the recorded data and component replacement events until that period. Then, we evaluate in the following period(s).

The evaluation is based on the ability to predict a future component replacement. Since there is a certain degree of randomness in the real data, we wait for three predictions of "ALARM" before actually issuing an alarm and dispatching the technician. If a component replacement happens in the following 75 cycles, it is marked as a correct prediction. It will be considered an early alarm if a component replacement happens between 75 and 100 cycles. If a component replacement is performed without a previous alarm, it is a missed failure. Finally, if no component replacement is performed in the next 100 cycles, it is a false alarm. In a realistic scenario, when an alarm is issued, the technician is sent to check the status of the machine. If the technician finds that the machine is in a healthy state, we could consider the alarm to be false and the machine to be healthy for the near future. However, the models would likely keep issuing alarms, that could be ignored based on the expertise of the technician. For this reason, in our evaluation, we observe a cooldown period of 25 cycles after a false alarm.

4.4. Refinement of Component Selection

In the recorded sensor data, we can often observe many trends that turn out to be unrelated to the presence or absence of faults. For example, these trends typically relate to the external weather conditions, the temperature of the water, or the usage of the machines. The component replacement event selection from Subsection 4.2 can be very prone to picking up these spurious trends; since the number of examples is very low, it will therefore commonly lead to overfitting.

In order to avoid overfitting, we propose to add these false alarms as "soft labels" into the training process. The process is described in Figure 3. If we train a model with the data for some time, we will evaluate and identify the false alarms in the following testing period. For each of these false alarms, we will extract a data sample with the previous 150 cycles and add them to the selection process of Subsection 4.2. These data samples can be added to the training datasets, labeled as "SAFE." In practice, this means training models that try to predict as many faults as possible while keeping the number of false alarms low.

On the other hand, looking at Figure 2, it is natural to think that the difference in the number of maintenance operations between machines is not necessarily due to some machines being inherently better than the others. In other words, there is probably missing information in the maintenance history of some machines. This means that an unknown number of

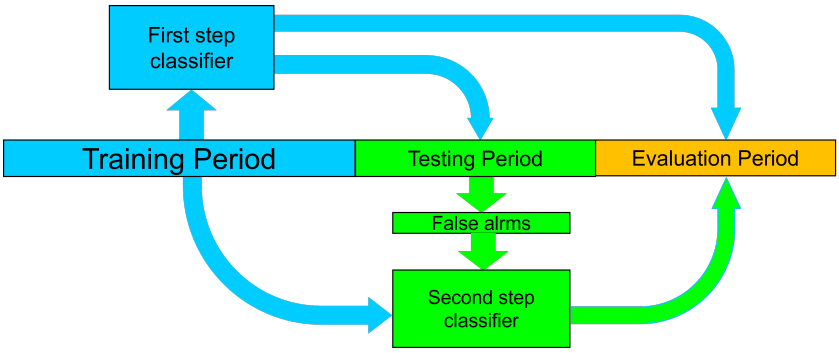

Figure 3. Workflow of the proposed methodology.

false alarms should be expected; that, in fact, they are not false alarms, and a fault might be happening. If we were to introduce the data samples for these alarms in the training process, we would be introducing confusing labeling again. Therefore, we need to select those false alarms' data samples.

We again use a genetic algorithm to select the component replacement events that are useful to predict each other. In addition, we add the false alarms events that are not predicted as "ALARM."

Algorithm 2 Fitness function for the refinement step.

N set of component replacement events and false alarm events.
$D_i = [(\mathbf{x^1}, y^1)...(\mathbf{x^m}, y^m)]$ dataset for event i
R vector of results
for $i = 1$ to $length(N)$ **do**
    $D_{train} = \bigcup_{j \neq i} D_j$
    $m = \text{trainclassifier}(D)$
    $\mathbf{pred} = \text{predictprobability}(D_i)$
    $R_i = \text{measureAUC}(\mathbf{pred}, \mathbf{y_i})$
end for
F fitness value
for $r = 1$ to $length(N)$ **do**
    if i corresponds to a component event **then**
        if $R_i > threshold_1$ **then**
            $F = F + 1$
        else
            $F = F - 1/3$
        end if
    else
        if $R_i < threshold_2$ **then**
            $F = F + 1/10$
        end if
    end if
end for

Now the fitness function in Algorithm 2 accounts for the number of component replacements that are correctly predicted. There is also a bonus for the number of false alarm events that are not predicted as "ALARM."

5. Experiments and Results

Our goal in the following experiments is to compare the performance of the classifier trained on the selected component replacement events as described in Subsection 4.2 (from now on, First Step Model) and the performance of the classifier trained using the selected component replacement events and the false alarms created by the first model, as described in Subsection 4.4 (from now on, Second Step Model).

To do so, we train a First Step Model for a given period n. We use the period $n + 1$, to evaluate the presence of false alarms. We then use those false alarms to train a Second Step Model on the component replacement events of period n, and the false alarms created during period $n+1$. We will compare both models on period $n + 2$ and the following. In practice, we have two years of data and four periods, so this means two comparisons.

5.1. Training During Period 1

In total, there are 34 component replacements in period 1. To select the fault component replacement events, we choose a threshold for the area under the ROC curve of 0.9, a very restrictive value.

Table 1. Results of predicting failures during periods 3 & 4, based on classifiers trained on period 1 (First Step) or periods 1+2 (Second Step). Comparison based on the Correctly Predicted Replaced Components (CPC), Early Alarms (EA), Missed Component Replacement (MCR) and False Alarms (FA).

	CPRC	EA	MCR	FA
First Step	24	0	70	76
Second Step	30	0	64	67

In selecting the component replacement events useful to predict each other, the genetic algorithm chooses 17 component replacement events, among which only 7 are predicted with more than 0.9 of area under the ROC curve. This means that only about 20% of the component replacement events are selected.

For the Second Step Classifier, we use the 34 component replacements during period 1, and the 76 false alarms from period 2. After the genetic algorithm performs the selection, 18 component replacement events were selected, among which only 5 had a predicted area under the ROC curve bigger than 0.9. In addition, 25 false alarms were selected.

The results of both approaches evaluated on periods 3 and 4 can be seen in Table 1. Not only is the Second Step reducing the number of false alarms by more than 10% (keep in mind that an unknown number of false alarms is to be expected); but we have also increased the number of correctly predicted component replacements by 25%.

There are many missed component replacements. As an indication, though, we should keep in mind that we used 20-25% of the component replacement events for training. This value roughly coincides with the ratio of correctly predicted component replacement to missed component replacements.

5.2. Training During Periods 1 and 2.

In total, there are 98 component replacements during periods 1 and 2. To select the fault component replacement events, we use again the threshold for the area under the ROC curve of 0.9.

Table 2. Results of predicting failures during period 4, based on classifiers trained on periods 1+2 (First Step) or periods 1+2+3 (Second Step). Comparison based on the Correctly Predicted Replaced Components (CPRC), Early Alarms (EA), Missed Component Replacement (MCR) and False Alarms (FA).

	CPRC	EA	MCR	FA
First Step	7	0	40	53
Second Step	12	0	35	47

After selecting the component replacement events that are useful to predict each other, the genetic algorithm selects 53 component replacement events, among which only 26 are predicted with more than 0.9 of area under the ROC curve. This means that only about 25% of the component replacement events are selected.

For the Second Step Classifier, we use the 98 component replacements during periods 1 & 2 and the 53 false alarms from period 3. After the selection performed by the genetic algorithms, 51 component replacement events were selected, among which only 21 had a predicted area under the ROC curve bigger than 0.9. In addition, 11 false alarms were selected.

The results of both approaches evaluated on period 4 can be seen in Table 2. Like in the previous experiment, the two-step approach has significantly increased the number of correctly predicted component replacements and the final number of false alarms has been decreased.

The ratio of correctly predicted component replacements and missed component replacements is just 15%, compared to the 25% of component replacement events selected in the training phase. For the second step classifier, this ratio is about 20% in the selection phase and 25% in the evaluation phase.

6. Conclusions and Future Work

In this paper, we have proposed a methodology to deal with the problem of misleading repair logs that can be harmful when creating machine learning-based predictive models. Using this misleading information to train models for predictive maintenance often leads to poor performance in practical settings when the quality of available data is not very high.

There are multiple reasons for this misleading information existing in reality. We can summarize them as follows: we cannot be sure that the replacement of a component in a machine is caused by the presence of a fault in said component; even if it is, we cannot be sure that reliable symptoms in the data exist to track the health deterioration; finally, we cannot be sure that no other maintenance operations have been performed in the machine, without being recorded in the service logs.

To deal with this problem, we first present a methodology to select those component replacements that are useful to create good performance models. This somehow naive selection process has been demonstrated experimentally to necessarily lead to overfitting and a large number of false alarms when those models are used to predict future failures.

We further add a second step, proposing our new methodology, where false alarms created in the first step are used to refine our models. We expected to reduce the number of false alarms after the refinement phase, which we have achieved by a margin of more than 10%, as verified experimentally on real-world industrial data.

However, more interestingly, we have also improved the number of correctly predicted component replacements by a healthy margin of 25% or more. Adding the false alarms to the training phase not only reduces the number of predicted false alarms in the future but also improves the selection of component replacement events to create more accurate models.

Implicitly, we have assumed that there is just one fault mode in our machines. By simply selecting the component replacement events that are useful to predict each other, we expect our selection process to just focus on one type of fault. In reality, we know that a machine as complex as a sterilizer will have many different types of faults. A clear continuation of this work is to extend the methodology to accommodate different types of faults either by adapting the selection algorithm to naturally accommodate them or by performing the selection iteratively.

ACKNOWLEDGMENT

This work was partially supported by "Stiftelsen för kunskaps- och kompetensutveckling" and CHIST-ERA grant CHIST-ERA-19-XAI-012 funded by Swedish Research Council.

REFERENCES

Carvalho, T. P., Soares, F. A. A. M. N., Vita, R., da P. Francisco, R., Basto, J. P., & Alcalá, S. G. S. (2019). A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering*, *137*, 106024.

Frénay, B., & Verleysen, M. (2013). Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, *25*(5), 845–869.

Lei, Y., Yang, B., Jiang, X., Jia, F., Li, N., & Nandi, A. K. (2020). Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mechanical Systems and Signal Processing*, *138*, 106587.

Nguyen, D. T., Mummadi, C. K., Ngo, T. P. N., Nguyen, T. H. P., Beggel, L., & Brox, T. (2019). *Self: Learning to filter noisy labels with self-ensembling.* arXiv. Retrieved from `https://arxiv.org/abs/1910.01842` doi: 10.48550/ARXIV.1910.01842

Prytz, R., Nowaczyk, S., Rögnvaldsson, T., & Byttner, S. (2015). Predicting the need for vehicle compressor repairs using maintenance records and logged vehicle data. *Engineering applications of artificial intelligence*, *41*, 139–150.

Ricker, N. L. (1996). Decentralized control of the tennessee eastman challenge process. *Journal of process control*, *6*(4), 205–221.

Saxena, A., & .K, G. (2008). Turbofan engine degradation simulation data set.

Seale, M., Hines, A., Nabholz, G., Ruvinsky, A., Eslinger, O., Rigoni, N., & Vega-Maisonet, L. (2019). Approaches for using machine learning algorithms with large label sets for rotorcraft maintenance. In *2019 ieee aerospace conference* (pp. 1–8).

Sipos, R., Fradkin, D., Moerchen, F., & Wang, Z. (2014). Log-based predictive maintenance. In *Proceedings of the 20th acm sigkdd international conference on knowledge discovery and data mining* (pp. 1867–1876).

Thomas, D., & Weiss, B. (2020). Economics of manufacturing machinery maintenance: A survey and analysis of us costs and benefits.

Yang, D.-M., Stronach, A., MacConnell, P., & Penman, J. (2002). Third-order spectral techniques for the diagnosis of motor bearing condition using artificial neural networks. *Mechanical systems and signal processing*, *16*(2-3), 391–411.

BIOGRAPHIES

Pablo del Moral is a PhD candidate in Data Mining at Center for Applied Intelligent Systems Research, Halmstad University, Sweden. He has a Masters degree in Data Science from University of Granada, and a Masters degree in Nuclear, Particle and Astrophysics from Technical University of Munich.

Slawomir Nowaczyk is a Professor in Machine Learning, working at Center for Applied Intelligent Systems Research, Halmstad University, Sweden. He has received his MSc degree from Poznan University of Technology in 2002 and his PhD degree from Lund University of Technology in 2008. During the last decade his research focused on knowledge representation, data mining and self-organising systems, especially in large and distributed data streams, including unsupervised modelling. He is a board member for the Swedish AI Society, and a research leader for the School of Information Technology at the University of Halmstad. Slawomir has led multiple research projects related to applying Artificial Intelligence and Machine
Learning in many different domains, such as transport and automotive, energy, smart cities as well as healthcare. In most cases, this research was done in collaboration with industry and public administration organisations – inspired by practical challenges and leading to tangible results and deployed solutions.

Sepideh Pashami is a senior researcher at RISE and a lecturer at CAISR (Center for Applied Intelligent Systems Research) at Halmstad University. She received her Ph.D. degree from AASS Research Centre, Örebro University, Sweden, in 2016. Her research interests include predictive maintenance, interactive machine learning, causal inference, and representation learning. She has been involved as a researcher and research leader in many projects (e.g. EVE, In4Uptime, ARISE and HEALTH) together with
Volvo Group AB, applying machine learning for predictive maintenance of heavy-duty vehicles.