Fleet PHM for Critical Systems: Bi-level Deep Learning Approach for Fault Detection

Gabriel Michau[1], Thomas Palmé[2], and Olga Fink[3]

[1,3] *Zurich University of Applied Sciences, Rosenstr. 3, Winterthur, 8401, Switzerland*
gabriel.michau@zhaw.ch
olga.fink@zhaw.ch

[2] *General Electric (GE) Switzerland, Brown Boveri Str. 7, Baden, 5401, Switzerland*
thomas.palme@ge.com

ABSTRACT

Data-driven approaches are highly reliant on the representativeness of the dataset used for training the algorithms. For Prognostics and Health Management (PHM) applications, a lack of representativeness will result in detecting new operating conditions (that have not yet been observed during the period used for training) as faults. This is particularly a challenge for PHM in critical systems, for which long and consistent datasets with all operating conditions are generally not available.

Among the many data-driven approaches applied to PHM, Deep Learning has recently brought promising results, enabling an automation of traditional PHM tasks, including signal processing, feature engineering and signal monitoring. Yet, as the parameters to optimize in a Deep Neural Network are numerous, the training requires also huge datasets: another quality often missing in critical system datasets.

When a fleet of systems is monitored, however, a solution to compose more representative and bigger training datasets is to combine condition monitoring data from systems with similar operating conditions. Identifying similar operating conditions would, traditionally, require comparing the distances and the distributions of multi-dimensional time series, a computationally intensive task worsened by the curse of dimensionality.

In this paper, we propose to use a deep neural network, designed for measuring similarities between the training and the testing datasets: a Hierarchical Extreme Learning Machine (HELM). HELM has demonstrated excellent abilities to jointly learn features and monitor deviations from the training data. Training first this network on individual systems, HELM can be used to identify other systems with similar operating conditions. Afterwards, the same network can be trained again with this representative dataset composed of condition monitoring data of several systems, to monitor more efficiently the health of the individual systems. Any deviation in the algorithm output would signify that the system is not anymore in operating conditions seen in the training fleet, and is likely experiencing a fault.

The novelty of the proposed approach lies in the usage of the same architecture twice in a bi-level framework: first, for selecting the representative datasets from a fleet of systems and second, using the selected datasets to train the health monitoring algorithm. This approach achieves good performances on both tasks. Learning from the fleet attenuates the impact of changing operating conditions (e.g., summer/winter trends), and improves the reliability of the fault detection.

The approach is tested on a fleet of 112 power plants, some of which experienced a stator vane failure.

1. INTRODUCTION

1.1. Fleet Approaches for PHM Applications

The increased availability and decreased cost of condition monitoring data has fostered the application of data-driven PHM applications. One of their limitations in practical applications is the high dependence on the representativeness of data used for training the algorithms. A lack of representativeness of the training dataset will result in detecting operating conditions that have not yet been observed by the condition monitoring system as faults. Since the operating conditions of an asset can change over time, a particular care needs to be taken to distinguish between the evolving or not yet observed operating conditions and potential faults.

For condition monitoring systems that are newly taken into operation, collecting a representative dataset may require a substantial time period enabling to cover the specific operating and environmental conditions of the monitored asset. The longer it takes to collect a dataset with representative operating conditions, the longer it will take to provide reliable detection and diagnostics results. This delay may influence the acceptance of the PHM applications by their users in practice.

However, if not only one asset is monitored but a fleet of assets, the operational experience of several monitored assets can be exploited for composing a representative training dataset.

Fleets can generally be considered from two different perspectives: either that of an operator or that of a manufacturer. From the perspective of a single operator, a fleet of assets is defined as "a group of machines or assets organized and operated under the same ownership for a specific purpose" (Jin et al., 2015). The purpose of the assets makes them a fleet, not the operating conditions, but as they are used in similar ways, they are likely to have relatively homogeneous operating and environmental conditions.

In the context of this research, we choose the perspective of a manufacturer and define a fleet as a set of homogeneous systems with corresponding characteristics and features, operated under different conditions, not necessarily by a single operator (Giacomo Leone, Cristaldi, & Turrin, 2017). This broader perspective typically results in larger fleets with more variability in operating conditions and requires, for the fine analysis of the fleet, to form sub-fleet of similar operating systems. In this research, we consider a fleet of 112 gas turbines of a single manufacturer with similar configurations, operated under very different conditions and in different environments.

Several approaches have been proposed for fleet PHM (González-Pri Da et al., 2016; Lapira, 2012; Giacomo Leone et al., 2017; Liu & Zio, 2016; Zio & Di Maio, 2010). Most of the proposed approaches aim to transfer either the fault patterns or the degradation paths between the single assets of a fleet. Some of the proposed approaches aim at developing models valid at fleet level with subsequent adaptations to the specific operating conditions of a single asset (Liu & Zio, 2016).

One of the challenges to be solved for fleet PHM applications is typically the selection of similar assets within a fleet and thereby creating sub-fleets with a homogenous system behaviour. In some of the proposed applications, the similarity is assessed by diversity indices (González-Pri Da et al., 2016) based on the characteristics of operating hours, operating conditions, and usage profiles. However for large fleets, this approach may result in large sub-fleets and the selected characteristics may still not be sufficiently distinctive to compose a sub-fleet with homogeneous system behaviour.

The problem of sub-fleet selection is that of multi-dimensional time series comparison. However, most of the proposed approaches, have either focused on comparing one-dimensional usage and degradation patterns (G. Leone, Cristaldi, & Turrin, 2016), or alternatively on comparing similarities between single multi-dimensional condition monitoring measurements (Zio & Di Maio, 2010). In the first case, degradation rates for one selected parameter are computed, without taking multi-dimensional signals into consideration (Giacomo Leone et al., 2017), in the second case, time series information are not taken into consideration.

In Lapira, 2012, a two-step approach is proposed for sub-fleet selection: in the first step a global clustering is performed, in the second step a local clustering is performed based on a peer-to-peer comparison of the units.

Typically, selecting sub-fleets based on multi-dimensional condition monitoring time series requires comparing the distances and the distributions of multi-dimensional time series, a computationally intensive task worsened by the curse of dimensionality.

To overcome these challenges, we propose a bi-level deep learning approach for fault detection. The deep learning approach is based on a deep neural network which was designed to measure similarities between the training and the testing datasets: a Hierarchical Extreme Learning Machine (HELM). HELM has demonstrated excellent abilities to jointly learn features and monitor deviations from the training data (Michau, Palmé, & Fink, 2017). At the first level, the network is trained on individual systems. Designed to distinguish between similar and dissimilar operating conditions, it can be used to identify other assets within the fleet with similar operating conditions. Thereby, a sub-fleet with assets exhibiting the most similar system behaviour can be selected. At the second level, this enlarged dataset comprising the condition monitoring data of the sub-fleet is used to retrain the neural network to learn the representative operating conditions that are the most defining for the selected asset. Enlarging the operating experience of a single system to that of a sub-fleet in this way enables more efficient and more reliable health monitoring systems.

1.2. Introduction to the Stator Vane Case Study

The proposed approach is applied to a case study of a fleet comprising 112 power plants. While about 100 gas turbines have not experienced identifiable faults during the observation time period (approximatively one year), 12 units have experienced a stator vane failure.

A vane in a compressor redirects the gas between the blade rows, leading to an increase in pressure and temperature. The failure of a compressor vane in a gas turbine is usually due to a Foreign Object Damage (FOD) caused by a part loosening and travelling downstream, affecting subsequent compressor parts, the combustor or the turbine itself. Fatigue and impact from surge can also affect the vane geometry and shape and lead to this failure mode. This is particularly due to the fact that parts are stressed to their limits to achieve high operational efficiency with complex cooling schemes to avoid their melting, especially during high load.

Stator vane failures are undesirable due to the associated high costs, including repair costs and operational costs due to the unexpected shutdown of the power plants. Even though such mechanical failures are not frequent, they have severe consequences. Therefore, an early detection is a real challenge today.

So far, the detection of compressor vane failures mainly relies on analytics stemming from domain expertise. Yet, if the algorithms are particularly tuned for high detection rates, they often generate too many false alarms. False alarms are very costly, each raised alarm is manually verified by an expert which makes it a time- and resource-consuming task.

Because of the various different factors that can contribute to the source of the failure mode, including assembly, material errors, or the result of specific operation profiles, the occurrence of a specific failure mode is considered as being random. Therefore, the focus is nowadays on early detection and fault isolation and not on prediction.

However for the considered case study, due to the limited observation time period of one year, the gas turbines have not experienced all relevant operating conditions and simply training the algorithms on the condition monitoring data of a single turbine, will either result in false alarms or in missed detections. Applying the proposed bi-level approach enables to clearly separate between the new operating conditions that have not yet been observed by the asset and faulty system conditions.

The rest of the paper is organized as follows: Section 2 gives a first introduction to HELM neural networks. HELM is applied to individual turbines. Section 3 gives a detailed account of the fleet approach. The proposed approach is tested and quantified with the fleet of 112 turbines.

2. Individual HELM Approach

2.1. From Single Layer Neural Networks to HELM

Hierarchical Extreme Learning Machines are a deep version of Extreme Learning Machines. Extreme Learning Machines are a very specific kind of neural networks. They actually are very similar to single layer feed forward neural networks and take advantage of a mathematical proof stating that given enough neurons, any function can be approximated by a single layer neural network for which only the weights between the hidden layer and the output are learned. The weights between the inputs and the hidden neurons are drawn randomly. This has led to a broad interest in such random networks as the learning process is much easier than for traditional neural networks: it consists in minimising a relatively simple convex function. They are therefore very fast to train and easy to use.

This result is however *a priori* not useful for traditional deeper architecture. Applying a succession of randomly drawn weights is not likely to improve the accuracy on the output approximation. A solution to take advantage of Extreme Learning Machines in deeper architecture is the use of stacked architectures: it consists of a succession of unsupervised ELM (auto-encoders most of the time), where each hidden layer is used as the input of the next ELM, and of one last supervised ELM, trained to perform the task of interest.

In PHM, such Hierarchical architectures (HELM) are attractive. By training successive auto-encoders, the information contained in the signal is concentrated in hidden layers that could directly be interpreted as representative features. The last layer is, therefore, using these features to perform the task of interest. From a conceptual perspective it mimics, thus, the traditional PHM system design: feature engineering to have a concise description of the system and feature monitoring for identifying behavioural changes. Such framework for PHM is illustrated in Figure 1.

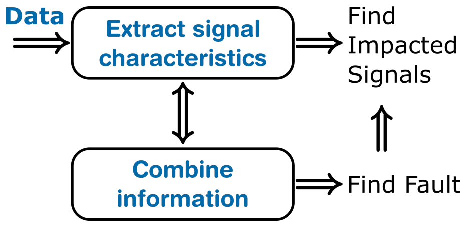


Figure 1. **PHM framework for Fault Detection and Isolation**. It consists of a first layer for feature learning and of a second layer, using the feature for health monitoring. Features are used for identification once abnormalities have been detected.

2.2. Health Indicator Monitoring Framework based on One Class Classifier

Among the many challenges in applying machine learning to PHM problems, one of the most common is the unbalanced dataset problems. There is very often a lack of representative data for faults that one wishes to distinguish from the healthy system conditions and to distinguish between the different fault types. For complex and critical system it is particularly true and this is the consequence of several inherent particularities:

- Similar faults can impact the system differently depending on the operating conditions.
- Local faults can impact the whole system in unpredictable ways (chaotic consequences).

- For complex systems, one does not necessarily know all faults that can happen, much less observe them.
- For robust systems, faults are rare.
- For critical systems, faults cannot be afforded and are usually prevented with preventive maintenance operations. This results in a lack of data representative of faulty conditions.

Regular data-driven classification, as usually performed in machine learning tasks, is therefore not very adapted to the case of fault detection: the most naive model, which would classify everything as healthy would achieve almost perfect accuracy when faults are rare. Yet, the model would be useless. The one-class classifier is a solution to that problem: it is trained on a single class, which in this case, would be the healthy class. The output of the classifier is the confidence that the current data points belong to the healthy class, that is, to some extent the distance to the healthy class. It is conveniently interpretable as a Health Indicator. A validation dataset is used to estimate what fluctuations of that output are acceptable for healthy data points and thus helps in defining a threshold above which the data would be considered as abnormal. Experiments performed in (Michau, Yang, Palmé, & Fink, 2018) have shown that a good threshold is

$$\delta = 1.5 \cdot \text{percent}_{99}(|1 - Y_{\text{valid}}|) \tag{1}$$

where, percent_{99} is the 99th percentile function and Y_{valid} the output of the one-class classifier for the validation dataset.

2.3. The Integrated Fault Detection and Diagnostics Architecture

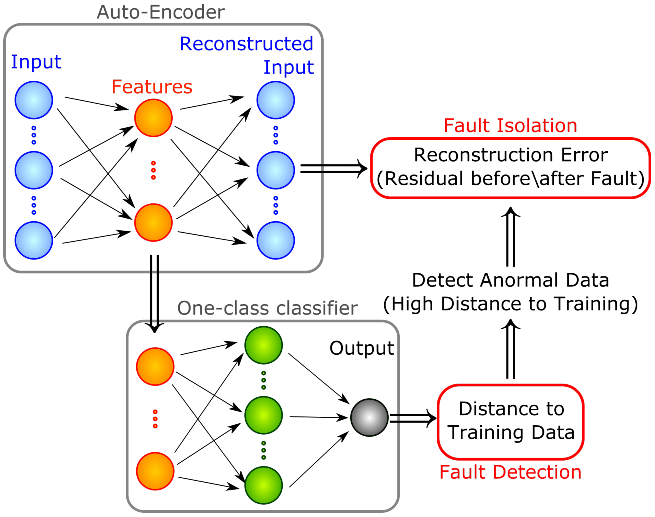

Auto-Encoder
Input
Reconstructed Input
Features
Fault Isolation
Reconstruction Error (Residual before\after Fault)
Detect Anormal Data (High Distance to Training)
One-class classifier
Output
Distance to Training Data
Fault Detection

Figure 2. **HELM architecture**. The HELM consists of an arbitrary number of stacked unsupervised auto-encoder ELM and of one last layer trained for the task at hand. In our case it is a one-class classifier ELM.

The complete architecture of the HELM is summarised in Figure 2. The auto-encoders ELM are trained with a regularisation term encouraging sparse connections (ℓ_1-regularisation). The idea is to use as few features as possible to reconstruct each dimension of the input (maximising information provided by the features). The one-class classifier is trained with a more traditional ℓ_2-regularisation preventing over-fitting only.

More details on the HELM equations, training and testing algorithm can be found in (Michau et al., 2017; Michau et al., 2018). Extensive experiments have been conducted to demonstrate the good abilities in fault detection, and in fault isolation, compared to traditional tools used in PHM applications.

2.4. Application of the Single Asset Approach to a Fleet of Gas Turbines

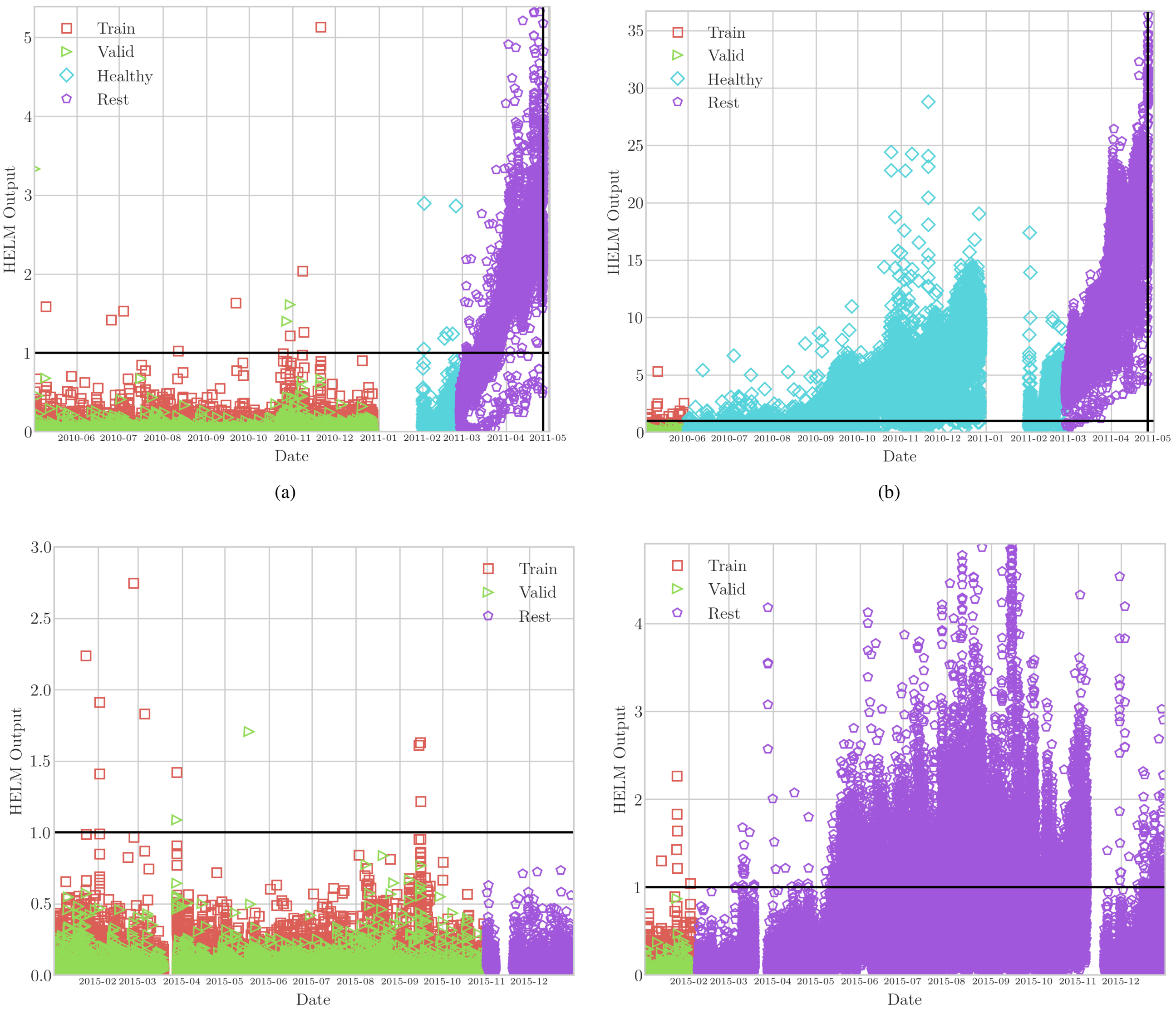
The one-class classifier HELM has been tested on a set of 112 datasets gathered on stator some of which experienced a fault.

Each dataset consists of nine variables sampled every five minutes for approximatively one year. These variables are ISO variables, that is that they correspond to measured variables, modified by a model, to the value they should have had in standard operating conditions (15^{o}C, 1 atmosphere).

The datasets are cleansed from missing or nonrealistic values (negative and null) and from values measured when the turbine was in a non-running or unstable state (e.g., , startup and shutdown). Each dataset is also rescaled such that the variables are centred and that the first and 99th percentiles would be -1 and 1 respectively. Then each dataset is split in two parts: a first part from which training and validation points are randomly sampled and a second part used for testing. For the datasets corresponding to “faulty” stators, the testing is again split in three: The data-points after the detection of the fault by the experts are labelled as unhealthy and will be used for True Positive (TP) quantification. Up to two months before the fault, the data-points are labelled as “unknown”, as the fault could have started before it has been detected by the experts, and the data-points are not used for quantification. The remaining data-points are labelled as healthy and will be used to quantify the False Positives (FP).

By reducing the training and validation datasets to few thousand points only, it is possible to simulate the case of a newly installed stator, or a stator that experienced major maintenance: cases with a limited number of condition monitoring data samples and a limited number of observed operating conditions.

Figure 3 gathers some representative results of the HELM applied to individual turbines. More quantified results are presented in the following section for comparison with the fleet approach. Figure 3 illustrates the HELM output, that is, the absolute distance to the healthy class. The output has



(a)

(b)

(c)

(d)

Figure 3. **HELM Output**: (a) and (b) for a faulty dataset, (c) and (d) for a dataset without detected fault. (a) it looks like a fault is developing during the last two months, with maximal output value at detection time given by the expert (vertical black line). (b) with smaller training, most of the dataset is detected as abnormal, in particular in winter. (c) for the healthy sets, there are few false positive (few outliers) but (d) with smaller training (3 months), again, most of the dataset is detected as abnormal.

been rescaled such that the detection threshold as per Equation (1) is 1. The two upper sub-figures 3a and 3b represent the HELM output for an "unhealthy" set, where the HELM has been trained with 80 000 and 12 000 points respectively. In the first case, the HELM is trained with as many healthy data-points as possible (Fig. 3a) and it looks like, in the last two months, the turbine behaviour is diverging from the learned system behaviour. With only three months of training (Fig. 3b), most of the dataset is detected as abnormal. Two peaks are particularly interesting to mention, the first one, around November to January 2011 corresponds to winter months, the second one, starting in march corresponds to what looked like a fault in Figure3a. Yet without the experiment with longer training, this potential fault would have been impossible to differentiate from changing operating conditions (winter), even when moving the detection thresholds. This problem of changing operating conditions is also illustrated with the "healthy" dataset experiments in Figures 3c and 3d. With long training, there are very few false positives in the testing, but with only three months of training, most of the dataset is detected as abnormal, with a peak centred 6 months after the last training points.

If with 10 months of training, faults could actually be detected for some of the faulty datasets, these results are unsatisfactory for mainly two reasons. First, all faults could not be detected, for some datasets the training was clearly not representative enough of the turbine behaviour. Second, the need for up to 10 months of training is, from the operator point of view, too long to wait for the implementation of a condition monitoring system. In addition, from this case study point of view, if more than 75% of the data is used for training, there is not much left to observe.

To address these limitations, we propose a fleet approach, that will use data from other turbines of a selected sub-fleet to constitute a training dataset adapted to the monitoring of each turbine.

3. Framework for Fleet PHM based on HELM

3.1. Motivation

One of the limitations of HELM, as it has been applied in the previous section, is the need of an extensive healthy dataset representative of all operating conditions in order to train the HELM. It is in fact a problem for most machine learning approaches to be able to perform detection on a new system, a system for which not much data are available or a system operated under new conditions.

As such, using data stemming from many similar systems (a fleet) seems a natural way to expand the training dataset. Yet, within fleets, systems can operate quite differently. The systems could have different configurations, the operating conditions could be different (different location, different environment, different requirements), the systems could also be of different age, with different maintenance history. Some parts might also have changed or been replaced with newer components. For the preparation of a good training dataset, it is therefore important to identify similar systems: if the system behave similarly, then we can assume their data could be combined for the training of a more robust model.

To achieve this objective, we propose here to use the HELM as a similarity measurement tool. As presented above, the HELM output is actually a value that can be directly interpreted as a distance between tested data points and the training dataset. We propose here to take advantage of that interpretation of the HELM output to identify similar datasets: Two systems are assumed to be similar if an HELM trained with one and tested on the other outputs low values.

3.2. HELM as Similarity Measure

In the fleet of gas turbines case study, for each dataset i we applied the following methodology:

1. To simulate a system with limited data availability (e.g., a recently installed turbine), we select the first three months of data in i. These three months are compared to all other healthy datasets in a two-way comparison:
   (a) A first HELM is trained[1] with data from the dataset of interest, i, and tested on every healthy dataset h_j, $j \in N$, where N is the number of healthy datasets in the case study. The ratio $r_i^{h_j}$ of points in h_j above threshold is computed.
   (b) Then, for each healthy dataset h_j, an HELM is trained[1] on h_j, randomly selecting a number of points corresponding to three months of data (but without temporal constraints). The HELM is tested on the data selected in i. Similarly, the ratio $r_{h_j}^i$ of points in i above threshold is computed.
2. The average
$$d = \frac{r_i^{h_j} + r_{h_j}^i}{2} \tag{2}$$
is the dissimilarity measure between i and h_j. The datasets h_j for which d is below a given threshold are selected as similar datasets to i.

Overall, the assessment of the fleet of turbines reveals that the fleet has a very high variability and each turbine has very strong specificities (even if the measured variables are converted to ISO conditions). The threshold on d needs to be above 30% in order to have 90% of the datasets with at least one similar turbine, as illustrated in Figure 4. Figure 4 represents the number of datasets with at least one other similar one as a function of the threshold on the similarity measure d.

[1] When training an HELM, 95% of the data available is used for training and the 5% remaining are used for validation (particularly for computing the threshold as per Equation (1)).

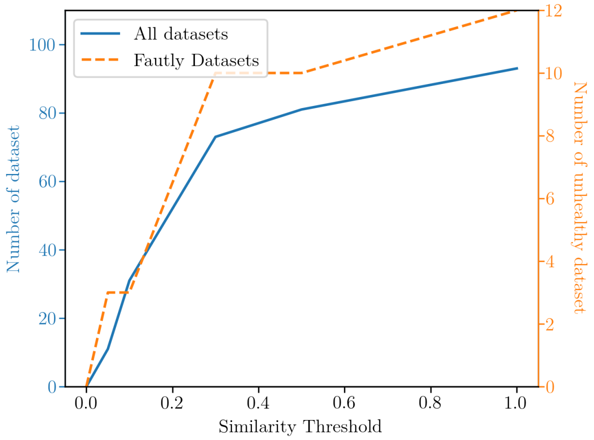


Figure 4. **Dataset Similarities**. For each dataset pair in the case study, mutual HELM are trained and tested on each other. The ratio of points detected as abnormal gives the dissimilarity. A threshold on this dissimilarity defines “similar” datasets. The number of datasets with at least one similar dataset is plotted as a function of the threshold for the whole fleet (plain blue) and for “faulty” datasets (dashed orange).

This means that in most cases, a large portion of the testing datasets are above threshold, but this is *a priori* not a bad thing: first it could have been expected that, with only three month of training, the HELM output still has a lot of variability. Second, we look for datasets that are similar but which could also bring new information to the training dataset for slightly different operating conditions. Thus, a relatively low number of points above threshold means that even if the datasets are quite similar, some so far unseen operating conditions are present in the testing dataset. The underlying idea is that by including those operating conditions in the training dataset, the representativeness and the robustness of the trained model will be adequately enlarged.

3.3. Results of the Proposed Fleet PHM Framework (HELM Trained on Similar Sub-fleets)

Once, for a turbine of interest, similar turbines have been identified with the above-presented methodology, we sample randomly training points from each similar turbine, including the first three months of the turbine of interest. This constitutes the training dataset of our final monitoring HELM for this specific turbine.

Once trained, the HELM can be tested on the remaining part of the dataset, that is, the next 9 months. For unhealthy turbines, this consists of healthy, unknown and unhealthy state data. Healthy and unhealthy state data are used for the computation of false positives and true positives, while unknown state data are let for future analysis (e.g., early detection). For healthy turbines, the remaining data is in healthy state only.

Figure 5a illustrates the impact over the fleet of this approach. Three indicators are presented in the Figure:

1. the ratio of datasets for which the number of False Positives has decreased

2. this same ratio for unhealthy datasets only

3. the ratio of datasets for which the number of True Positives has increased.

It should be pointed out that for any threshold and for almost 100% of the datasets, the number of false positives decreases, which is currently one of the major concerns in gas turbine condition monitoring. In addition, for a threshold up to 0.3, more than half of the unhealthy datasets have increased number of true positives. Yet, when the threshold increases, the method is less and less beneficial. This behaviour is expected: a higher threshold corresponds to less similar datasets and indicates a training dataset that is less and less specific to the turbine of interest. The training consists of operating conditions further and further away from the one of the turbine of interest, which increases the likelihood of missing major changes in behaviour.

In Figure 5b, the average ratio of removed false and true positives against the individual approach is plotted against the similarity threshold. Overall, using the fleet approach removes in average 80% of the false positives, a very encouraging result. The number of true positives decreases on average. If this is, *a priori* not a positive result, this is mitigated by two points: First, in the individual approach, as seen in Figure 3, most of the testing is above threshold. The ratio of true positives might be high, but these are not useful true positives as it is dwarfed by the number of false positives. With the fleet approach, by reducing the number of false positives by more than 80%, the meaning of true positives is reinforced. Second, it is normal that, in the fleet approach, as the training dataset fluctuates more and is representative of more operating conditions, the number of points above threshold diminishes both for false and true positives. Yet, it is important to notice in Figure 5b, that the decrease in true positives is far below that of the false positives.

Combining all these results, taking a similarity threshold at 0.3 appears to be a good compromise between number of datasets with at least one similar other turbine, the reduced number of false positives and the number of true positives. Another choice could have been 0.1, which is much better in terms of true positive statistics but it has not been chosen here due to the small number of datasets with at least one similar turbine. For the same dataset as that presented in Figure 3, the results of the fleet approach are presented in Figure 6. In Figure 7, the results of two additional turbines, one “faulty” and one healthy, are presented. Very similar as results can be drawn. Yearly changes impact the results from individually trained HELM but are mitigated by the fleet approach.

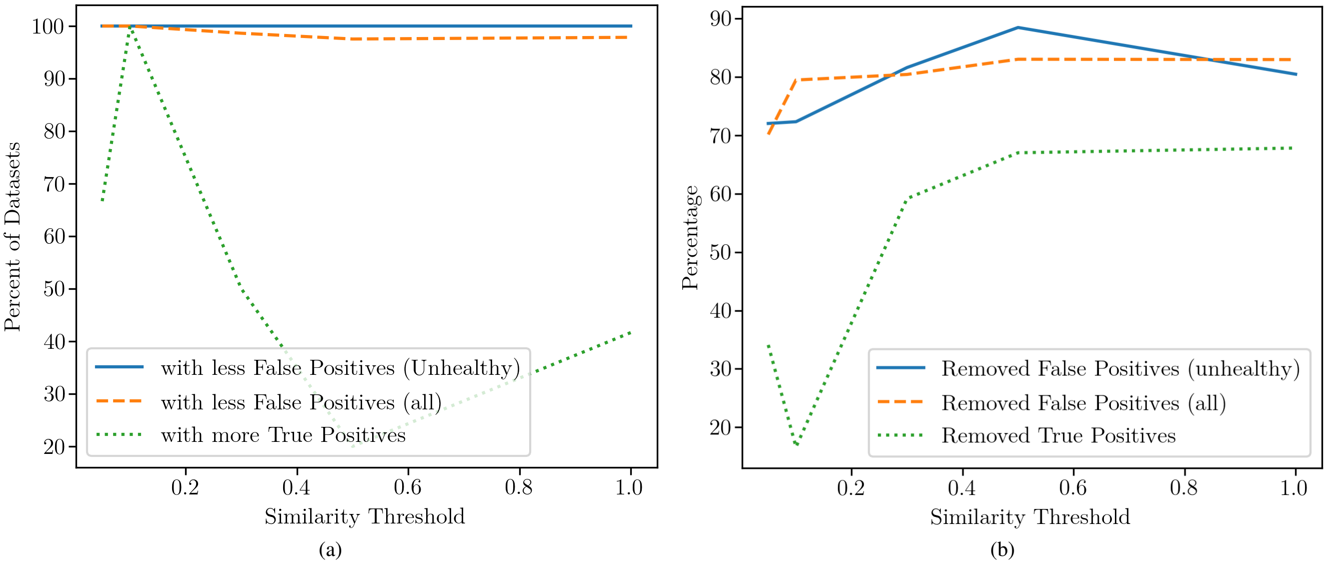


Figure 5. **Fleet Approach: FP and TP**. Applying the fleet approach, (a) the ratio of dataset for which the number of false positive decreased is plotted against the similarity threshold in plain blue. For unhealthy datasets, this same ratio is plotted in dashed orange and the ratio of datasets with more true positives is plotted in dotted green. (b) Removed FP and TP when applying the fleet approach as a function of the similarity threshold. Applying the fleet approach removes around 80% of the false positives. It removes also some true positives but in smaller proportion, giving stronger meaning to the true positives that are left.

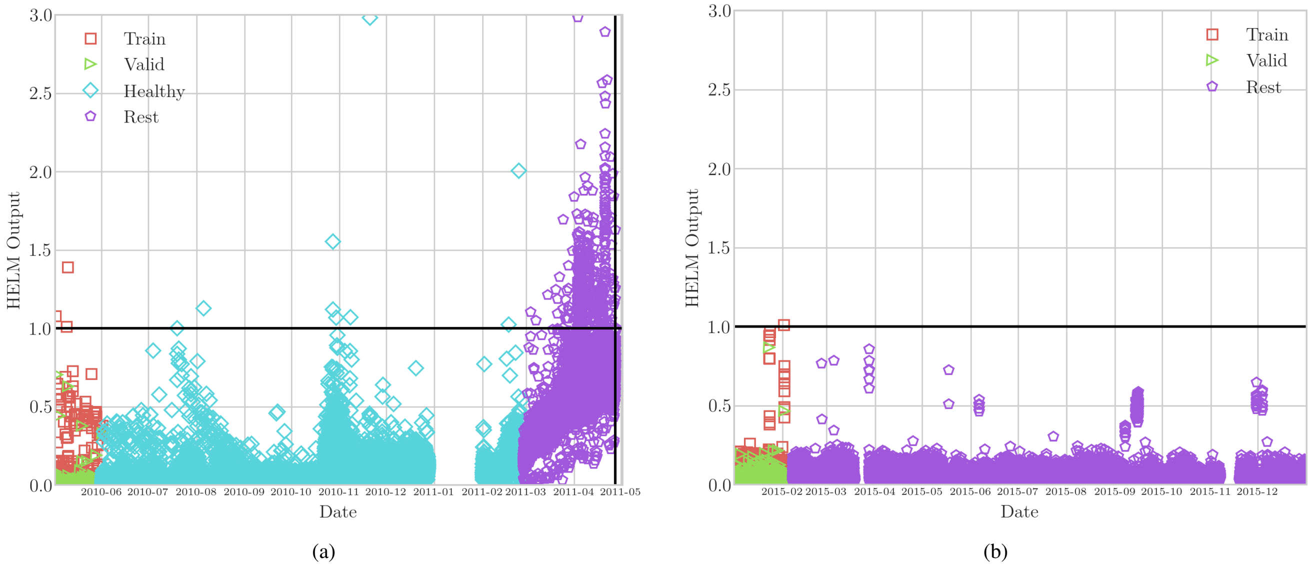
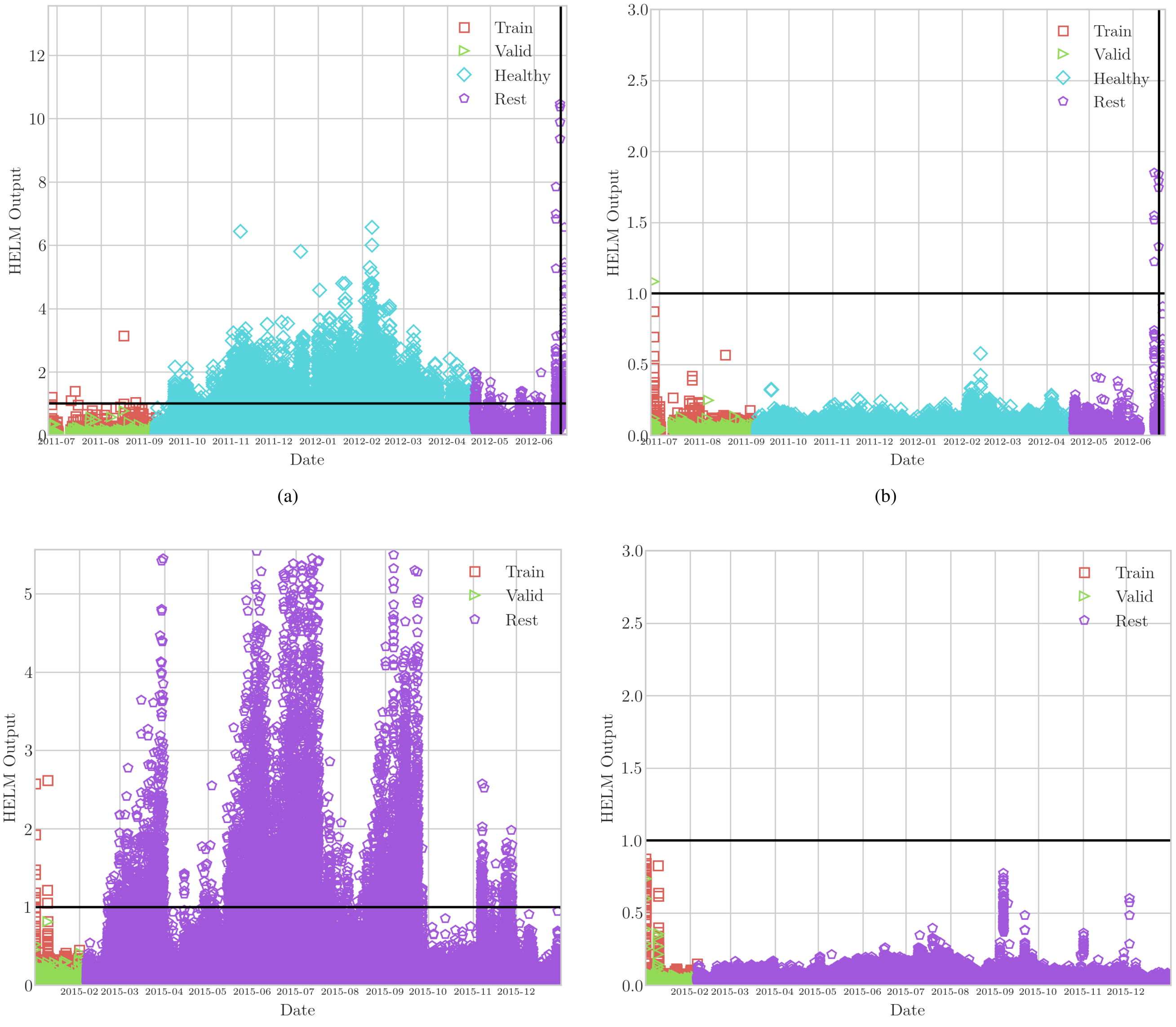


Figure 6. **HELM Output**: (a) for the faulty dataset, (b) for the dataset without detected fault. (a) The fleet approach confirms the developing fault as observed first in Figure 3a, with a maximum output value at detection time given by the expert (black vertical line). (b), the dataset is now detected as healthy.



(a)

(b)

(c)

(d)

Figure 7. **HELM Output**: (a) and (b) for a faulty dataset. The HELM has been trained (a) on three months of data, (b) with the fleet approach. Similarly as in Figure 3a, winter conditions are detected as unhealthy in the individual approach, but not in the fleet approach. A fault is confirmed at detection time given by the expert (vertical black line). (c) and (d) for an healthy turbine. The HELM has been trained (c) on three months of data, (d) with the fleet approach. Similar conclusion can be drawn as in Figure 6b.

4. CONCLUSION

In this paper, we presented a bi-level framework for a fleet approach in PHM. Fleet approach is challenging as it requires the combination of two difficult tasks: First, identifying a sub-fleet of assets with similar characteristics. Second, training a condition monitoring model for each system. The representativeness of the training data for the different operating conditions is crucial, while improper training would increase the likelihood of missing important changes in the system behaviour. We proposed a single tool to solve these two problems in a bi-level approach based on the recent encouraging results obtained with HELM. This approach, however, is technologically agnostic and could be applied with other health monitoring approaches. First the health monitoring tool is used on pair of systems: if they mutually detect each other as healthy, then they are likely to be similar. Then, these similar systems are used to train the health monitoring tool again and to monitor individual systems.

We demonstrated here that such approach greatly improves the results compared to the approach based on individual training, in particular with strong data constraints on the history of the system. The fleet approach can be used either for systems newly taken into operation or newly equipped with a condition monitoring system and is highly beneficial in the case of evolving fleets. If the systems are subject to regular maintenance or if the fleet changes often, historic fleet data can help to improve the monitoring of individual systems.

Yet, particularly in our case study, the fleet approach is still not enough for the condition monitoring of every single turbine. For a non-negligible number of cases, similar turbines could not be identified while the individual approach was also not working. Consequently, the choice of the similarity threshold is actually crucial. If too low, the method is efficient but only for the few systems that are extremely close. If set higher, more systems will be similar but the fleet-training data will be less representative of the system of interest specificity and might miss some important changes. In future research, this point might be solved by changing the way similar data are found. Rather than looking for other turbines whose entire dataset is similar, one could try to iteratively find small batches of data from the whole fleet that would be beneficial for the training dataset.

ACKNOWLEDGEMENT

This research was funded by the Swiss Commission for Technology and Innovation under Grant no. 17833.2 PFES-ES.

REFERENCES

González-Pri Da, V., Orchard, M., Martí, C., Guillén, A., Shambhu, J., & Shariff, S. (2016). Case Study based on Inequality Indices for the Assessments of Industrial Fleets, 250–255.

Jin, C., Djurdjanovic, D., Ardakani, H. D., Wang, K., Buzza, M., Begheri, B., … Lee, J. (2015). A comprehensive framework of factory-to-factory dynamic fleet-level prognostics and operation management for geographically distributed assets. In *2015 ieee international conference on automation science and engineering (case)* (pp. 225–230).

Lapira, E. R. (2012). *Fault detection in a network of similar machines using clustering approach* (Doctoral dissertation, University of Cincinnati).

Leone, G. [G.], Cristaldi, L., & Turrin, S. (2016). A data-driven prognostic approach based on sub-fleet knowledge extraction. In *14th imeko tc10 workshop on technical diagnostics 2016: New perspectives in measurements, tools and techniques for systems reliability, maintainability and safety*.

Leone, G. [Giacomo], Cristaldi, L., & Turrin, S. (2017). A data-driven prognostic approach based on statistical similarity: An application to industrial circuit breakers. *Measurement*, *108*, 163–170.

Liu, J., & Zio, E. (2016). A framework for asset prognostics from fleet data. In *2016 prognostics and system health management conference (phm-chengdu)* (pp. 1–5).

Michau, G., Palmé, T., & Fink, O. (2017, October). Deep Feature Learning Network for Fault Detection and Isolation. In *Annual Conference of the Prognostics and Health Management Society 2017*. Annual Conference of the Prognostics and Health Management Society 2017, St. Petersburg, Florida.

Michau, G., Yang, H., Palmé, T., & Fink, O. (2018, February). Feature learning for fault detection in high-dimensional condition-monitoring signals. *submitted*, 1–10.

Zio, E., & Di Maio, F. (2010). A data-driven fuzzy approach for predicting the remaining useful life in dynamic failure scenarios of a nuclear system. *Reliability Engineering & System Safety*, *95*(1), 49–57.