Hybrid deep fault detection and isolation: Combining deep neural networks and system performance models

Manuel Arias Chao[1], Chetan Kulkarni[2], Kai Goebel[3] and Olga Fink[1]

[1] *Chair of Intelligent Maintenance Systems, ETH Zurich, Switzerland*
{manuel.arias, ofink}@ethz.ch

[2] *SGT, Inc., NASA Ames Research Center, USA*
chetan.s.kulkarni@nasa.gov

[3] *Luleå University of Technology, Operation and Maintenance Engineering, Luleå, Sweden*
kai.goebel@ltu.se

ABSTRACT

With the increased availability of condition monitoring data on the one hand and the increased complexity of explicit system physics-based models on the other hand, the application of data-driven approaches for fault detection and isolation has recently grown. While detection accuracy of such approaches is generally very good, their performance on fault isolation often suffers from the fact that fault conditions affect a large portion of the measured signals thereby masking the fault source. To overcome this limitation, we propose a hybrid approach combining physical performance models with deep learning algorithms. Unobserved process variables are inferred with a physics-based performance model to enhance the input space of a data-driven diagnostics model. The resulting increased input space gains representation power enabling more accurate fault detection and isolation.

To validate the effectiveness of the proposed method, we generate a condition monitoring dataset of an advanced gas turbine during flight conditions under healthy and four faulty operative conditions based on the Aero-Propulsion System Simulation (C-MAPSS) dynamical model. We evaluate the performance of the proposed hybrid methodology in combination with two different deep learning algorithms: deep feed forward neural networks and Variational Autoencoders, both of which demonstrate a significant improvement when applied within the hybrid fault detection and diagnostics framework.

The proposed method is able to outperform pure data-driven solutions, particularly for systems with a high variability of

operating conditions. It provides superior results both for fault detection as well as for fault isolation. For the fault isolation task, it overcomes the smearing effect that is commonly observed in pure data-driven approaches and enables a precise isolation of the affected signal. We also demonstrate that deep learning algorithms provide a better performance on the fault detection task compared to the traditional machine learning algorithms.

1. INTRODUCTION

Increasing amounts of condition monitoring (CM) data from complex engineered systems, both in terms of the number of sensors as well as in terms of the sampling frequency, and advancements in machine and deep learning algorithms provide an untapped potential to extract information on asset health condition. Concretely, deep learning algorithms have demonstrated an excellent ability to learn the system behaviour directly from large volumes and variety of the condition monitoring signals and therefore decreased the need of manual feature engineering. As a result, deep learning-based solutions have been increasingly applied to complex learning tasks in prognostics and system health management (PHM) of complex systems (R. Zhao et al., 2016; Khan & Yairi, 2018).

Since machine and deep learning algorithm rely on learning patterns from representative examples, one of the major challenges in applying deep learning algorithm for fault detection and diagnostics tasks is the lack of labeled data, i.e. a lack of a sufficient number of representative samples of known fault patterns. Only a representative dataset of possible fault types would enable the algorithms to learn all the characteristic patterns of the specific faults and provide very good fault detection and isolation capabilities. Because faults are rare in complex safety critical systems, such as aviation propulsion

systems, it is unfeasible to obtain sufficient samples from all possible fault types that can potentially occur. However, most of the previous research in fault detection and diagnostics has been focusing on defining the problem of fault detection and diagnosis as a classification task and rather tackling the problem of imbalanced datasets for the faulty classes (S. Wang, Minku, & Yao, 2013; Xu, Chow, & Taylor, 2007; Zhang, Li, Gao, Wang, & Wen, 2018).

In this paper, we consider the case where we have only information on the healthy class, the number and nature of the fault classes are, however, not known in advance. This is a more realistic task for the practical applications but also a more difficult task compared to the case where the available labeled data samples already cover the essential information on the number and type of classes and the new observation only fall in the category of already known classes. As an additional degree of difficulty, we focus particularly on systems that are operated under varying conditions that are frequently changing. An example for such systems are aircraft engines experiencing continuous changes (transients) on the flight conditions.

One of the previously most common approaches for the case when only healthy system conditions are available for model development is based on signal reconstruction and the subsequent analysis of the residuals between the monitored and reconstructed signals (Baraldi, Di Maio, Turati, & Zio, 2015; Hu, Palmé, & Fink, 2017). Robust decision boundaries are crucial in this case for the performance of the algorithms. If condition monitoring signals are highly correlated, a so called smearing effect can occur influencing not only signals directly affected by the fault but also causing deviations in correlated signals that do not contain any information on the fault. This effect makes it difficult or even impossible to isolate the root cause of the fault if the fault isolation is solely based on the residuals (Hu et al., 2017).

Recently, a new integrated fault diagnosis approach was introduced, combining feature learning with a one-class classification for the fault detection and a subsequent analysis of the residuals for the fault isolation task (Michau, Palmé, & Fink, 2017). This solution strategy aims to map the observed healthy operation to a healthy class and later discriminate if the operating condition of interest with unknown health state follows the learned pattern of the healthy system conditions. The detection accuracy of such approaches is generally very good when the available healthy operating conditions used for training are clearly representative of the conditions under analysis.

Varying operating conditions create a shift in the underlying distributions of the CM data. Training an algorithm on the combined representation of these operating conditions with a limited number of samples may result in an unsatisfactory performance of the algorithms since the fault characteristics may be masked by the variability of operating conditions.

If the operating conditions are too dissimilar, a possible way to address this challenge would be to develop dedicated algorithms for each of the operating conditions and switch between the different algorithms depending on the operating condition of the current observation. Another way to benefit from the experience of several operating conditions is to apply domain adaptation and align the underlying distributions in the feature space (Q. Wang, Michau, & Fink, 2019), enabling thereby the transfer of the experience between the different operating conditions. However, such alignment requires at least some labels in the training dataset for the fault types which we don't have for the selected problem setup.

In this work, we focus on the challenging problem of fault detection and diagnostics under varying operating conditions and highly correlated signals. We propose a framework and a method that combines physics-based models and deep learning algorithms and is particularly targeting the case when faulty samples are not available during model development.

Complex systems can be modelled at various levels of detail, ranging from simple algebraic relations to full 3D-description of the process. In this range, thermodynamic models (a.k.a. 0-D models) of different levels of fidelity are generally available for design or control of complex systems. These models typically a moderate computational load and yet are able to predict process measurements (e.g., temperatures, pressures, air mass flow rates, rotational speeds) as well as global system performance (e.g. efficiencies and power). Furthermore, system performance models offer access to unmeasured variables that might be more sensitive to fault signatures and consequently can improve fault detection and isolation.

In the proposed framework, unobserved process variables are inferred with a physics-based performance model to enhance the input space of a data-driven diagnostics model. The resulting increased input space gains representation power enabling more accurate fault detection and isolation.

The focus of the proposed method is on fault detection and isolation for complex industrial assets that are operated under varying conditions. The main benefit of the proposed method arises particularly for systems for which we don't have sufficient labels to develop classification algorithms and for which pure data-driven approaches with a single model combining data from all the operating conditions provide unsatisfactory performance for fault detection and isolation.

The proposed hybrid framework can be combined with any deep learning algorithm. To demonstrate this, we combine it with a feed-forward neural network, a Variational Autoencoder and a vanilla autoencoder. To validate the fault detection and isolation capability of the proposed method, we generate a new dataset of an turbofan engine during flight conditions under healthy and four faulty operative conditions.

The dataset was synthetically generated with the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dynamical model (Frederick, Decastro, & Litt, 2007). Real flight conditions as recorded on board of a commercial jet were taken as input to the C-MAPSS model (*DASHlink - Flight Data For Tail 687*, 2012).The evaluated case study comprises simulated flight conditions under healthy and four faulty operating conditions.

To assess the effectiveness of the proposed hybrid framework, we first evaluate different deep learning architectures and then compare the performance to 1) pure deep learning algorithms with the same architecture as those applied within the hybrid approach; 2) a standard machine learning algorithm, the one-class support vector machines (OC-SVM) (Schölkopf, Williamson, Smola, Shawe-Taylor, & Piatt, 2000); 3) an alternative hybrid framework based on residuals between the performance model and the real observed condition monitoring data in combination with the real condition monitoring data as input.

We demonstrate that the proposed framework is able to outperform pure data-driven solutions, particularly for systems with a high variability of operating conditions. It provides superior results both for fault detection as well as for fault isolation. For the fault isolation task, it overcomes the smearing effect that is commonly observed in pure data-driven solutions and enables a precise isolation of the affected signal. We also demonstrate that deep learning algorithms provide a better performance on the fault detection task.

2. RELATED WORK

Data-driven and physics-based approaches have their advantages and limitations when applied as stand-alone approaches. While physics-based approaches do not require large amounts of data and retain the interpretability of a model, they are generally limited by their high complexity or incompleteness. On the contrary, data-driven approaches are simple to implement and are able to discover complex patterns from large volumes of data but are limited by the representativeness of the training datasets. The combined use of data-driven and physics-based approaches has the potential to lead to performance gains by leveraging the advantages of each method.

Different solutions have been proposed to combine physics-based models and data-driven algorithms. Depending on what type of information is processed and how the pieces of information are combined, different types of hybrid architectures can be created. In the following, some of the proposed architectures that are the most comparable to the proposed framework are presented and discussed.

Frank et al. (2016) explore the use of a hybrid approach where synthetic data of a healthy and faulty system are generated with a high-fidelity system model and used as input

to traditional data-driven algorithms. Within the hybrid architecture, a range of traditional data-driven machine learning algorithms with an additional feature engineering step was explored, including random forests and support vector machines. The output of the system model is subsequently combined with residuals between measurements and system performance estimation from a statistical model (i.e. generated based on historical data). This architecture (see Figure 1) is applied to diagnostics problems of abnormal energy consumption in buildings resulting from faulty equipment such as faults of air conditioners, chillers, dampers, and fan motors.
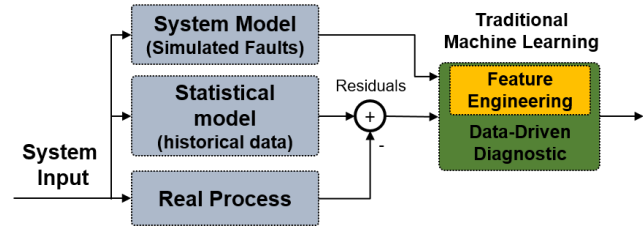

Figure 1. Overall architecture of the hybrid diagnostics approach in (Frank et al., 2016). The traditional machine learning algorithms take as input synthetic data of a healthy and faulty system that are generated with a system model and combined with the residuals between measurements and system performance estimation from a statistical model.

Hanachi et. al. (2017) use a parallel hybrid approach to diagnostics of gas turbines. In this approach, empirical (i.e. data-driven) fault transition models and physics-based system models perform the state assessment of the process at hand. Particle filter is used as a fusion mechanism to aggregate the diagnostic results from the measurement signals and degradation models (see Figure 2).
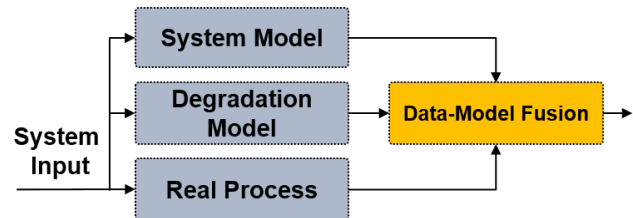

Figure 2. Overall architecture of the parallel hybrid diagnostics framework in (Hanachi et al., 2017). A particle filter is used as a fusion mechanism to aggregate the diagnostic results from measurement signals and degradation models.

A further possible architecture combining model-based and data-driven approaches uses first the system model to reason over the process and then a data-driven classifier that distinguishes between the different fault classes. Rausch et al. (2005) use such an architecture for online fault diagnostics of turbofan engines (see Figure 3). In their approach, features engineered from the residuals between estimates of Extended Kalman Filter (EKF) and sensor readings are used as input to a machine learning classifier (SVM model).
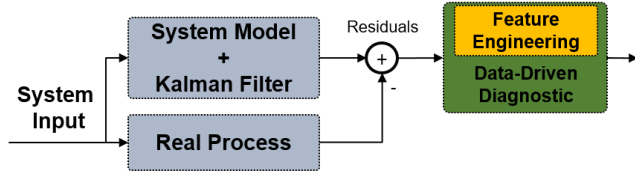
Figure 3. Overall architecture of the residual-based hybrid diagnostics in (Rausch et al., 2005). Feature engineering is carried out for the residuals between Kalman Filter estimates and sensor readings and are used as input to an SVM classifier.

Residual-based approaches are the standard for model-based diagnostics of aircraft engines. A generic residual-based diagnostics approach involves two major tasks. First, discrepancies between measurements and the expected healthy model responses are computed. In a second step, the residuals, that encode the potential the impact of degraded or faulty system behaviour, are processed with a fault detection and isolation (FDI) logic to create the diagnosis report (Borguet, 2012). Concretely, residuals can be fed as input to a deep-learning diagnostic algorithm in addition or instead of the measurements. Therefore, the fault detection and isolation logic is discovered by a deep neural network. Figure 4 shows a block diagram of a residual-based hybrid diagnostics framework where deep learning diagnostics algorithm receives as input the scenario-descriptor operating conditions and the residual between sensor readings and estimated model responses.
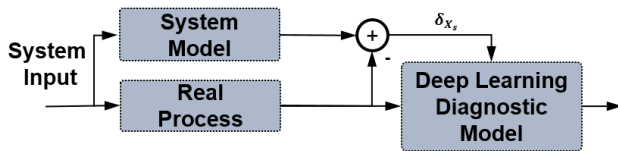

Figure 4. Overall architecture of the residual-based diagnostics approach. The deep learning diagnostics model receives as input the system inputs (i.e. scenario-descriptor operating conditions) and the residual between sensor readings and estimated model responses δ_{x_s}

Recently, several approaches of physics-guided machine learning have been proposed, where physical principles are used to inform the search of a physically meaningful and accurate machine learning model. The architecture proposed in (Jia et al., 2018), for example, enhances the input space to a data-driven system model with outputs from a physics-based system model. As a result, the dynamical behaviour of the system could be approximated more accurately.

In another variation of the physics-guided machine learning idea, a recurrent neural network cell was modified to incorporate the information from the system model at an internal state of the dynamical system (see Figure 5). A related idea was applied to a variety of prognostics problems, such as in (Nascimento & Viana, 2019; Dourado & Viana, 2019; Yuce-
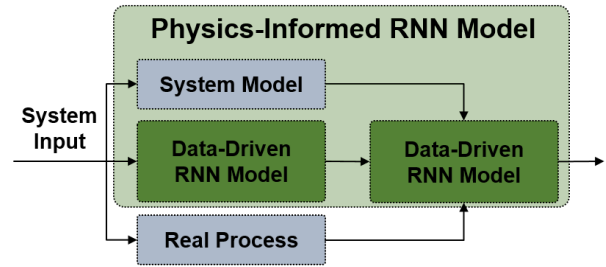
san & Viana, 2019).

Figure 5. Overall architecture of the physics-informed recurrent neural network in (Nascimento & Viana, 2019)

Contrary to previous hybrid architectures, the framework proposed in this paper leverages inferred **unobserved virtual sensors** and the **unobservable model parameters** to enhance the input space to a tailored deep learning-based FDI algorithm (see Figure 6).

3. PROPOSED METHODOLOGY

3.1. Calibration-Based Hybrid Diagnostics

Physics-based performance models of different levels of fidelity are generally available for design or control of complex engineered systems. System performance models are represented mathematically as coupled systems of nonlinear equations. The inputs of the model are divided into scenario-descriptor operating conditions w and model parameters θ. The output of the system model is not limited to estimates of measured physical properties values $\hat{x}_s$ but also provides unmeasured properties x_v (i.e. *virtual sensors*). As there is no description given by an explicit formula, the nonlinear performance model is denoted as

$$[\hat{x}_s, x_v] = S(w, \theta) \tag{1}$$

Performance models provide additional information that is not part of the condition monitoring signals and may be relevant for detecting faults. Therefore, we propose to make use of modelled variables $[\hat{x}_s, x_v, \theta]$ as input to the deep learning diagnostics algorithm. Hence, the resulting hybrid diagnostic approach combines information from physics-based models with CM data (i.e. $[w, x_s]$) and uses this enhanced input for subsequent fault detection and isolation with a deep learning algorithm.

However, to maximize the amount of relevant model information available for the generation of a data-driven diagnostics model, we propose to calibrate the system performance model $S(w, \theta)$. Model calibration involves inferring the values of the model parameters θ that make the system response to reproduce closely the observations x_s. Hence, the information about system degradation (and ideally the fault signature) is encoded within the estimated model correcting parameters $\hat{\theta}$. The calibrated model also provides high confi-

dence estimates of process variables $\hat{x}_v$ that may be sensitive to fault signatures. Therefore, we propose to enhance the input space for the deep-learning diagnostic model with the process variables $[\hat{x}_s, \hat{x}_v, \hat{\theta}]$ inferred with the system performance model. Figure 6 shows a block diagram of the proposed calibration-based hybrid diagnostic approach. The deep learning diagnostics model receives scenario-descriptor operating conditions w and model variables $[\hat{x}_s, \hat{x}_v, \hat{\theta}]$ as input. The feedback arrow to the system model represents the calibration process for updating the model calibration parameters $\hat{\theta}$. Model calibration is a standard approach in several technical areas including traditional model-based diagnostics (Brunell, Mathews, & Aditya Kumar, 2004), model-based control and performance analysis of system models (Arias Chao, Lilley, Mathé, & Schloßhauer, 2015).
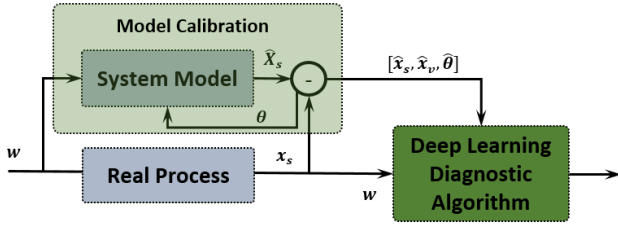

Figure 6. Overall architecture of the calibration-based hybrid diagnostics framework. The deep learning diagnostics algorithm takes as input the scenario-descriptor operating conditions w, estimates of the condition monitoring signals ($\hat{x}_s$) and the virtual sensors ($\hat{x}_v$) and model parameters (θ).

The extended representation provided by the calibrated system model also provides additional interpretability and ability to isolate potential degradation root causes. The model parameters θ are indeed model tuning of the system components and hence a deteriorated behaviour of a sub component is precisely encoded in only one component of θ (i.e. θ_k) while it is at the same time manifested in the condition monitoring data and virtual sensors. As it will be shown in the case study (Section 5), this feature avoids the smearing characteristic of data-driven diagnostics models. An additional advantage of including the calibration processing step is that errors in the sensor readings can be detected and removed and therefore diagnostics process is more robust to sensor faults.

In addition to the model calibration, a diagnostic report requires a clear fault detection and isolation algorithm, beyond the standard threshold-based logic. Therefore, we propose a tailored deep learning-based FDI algorithm shown in Figure 7. The proposed algorithm uses as input the extended representation provided by the calibrated system model ($x = [w, \hat{x}_s, \hat{x}_v, \theta]$) and computes a similarity score $s_I(x^{(j)}; \beta)$. Fault detection is performed based on a clear logic on $s_I(x^{(j)}; \beta)$. The enhanced input signal x is reconstructed with an autoencoder network and fault isolation is performed based on the similarity score $d_I(x_k^{(j)}; \nu_k)$. A detailed description of the
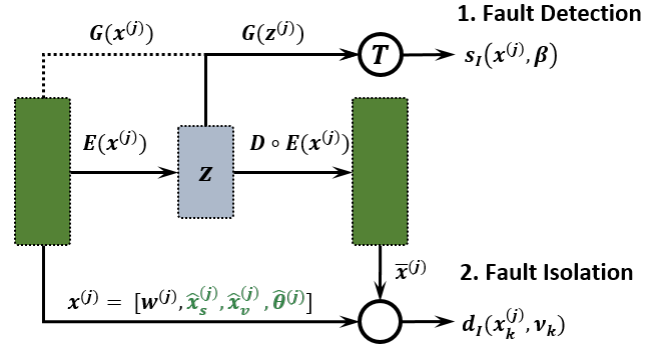
proposed algorithm is covered in Section 4.

Figure 7. Block diagram of the proposed fault detection and isolation algorithm within the proposes hybrid diagnostics framework. The FDI algorithm takes $x = [w, \hat{x}_s, \hat{x}_v, \theta]$ as input. A functional mapping $\mathcal{G}$ from the input (x) or an embedding representation of the input (z) to a target $\mathbf{T}$ is used to generate a similarity score $s_I(x^{(j)}; \beta)$. Fault detection is performed based on $s_I(x^{(j)}; \beta)$. The enhanced input signal is reconstructed ($\bar{(}(x))$) with an autoencoder network. Fault isolation is performed based on the similarity score $d_I(x_k^{(j)}; \nu_k)$. The variables in green are specific to the proposed hybrid approach.

4. METHODS

4.1. Problem Statement

We aim at developing a diagnostic model able to detect and isolate fault types on complex systems operated under a large range of changing operating conditions. In our problem, we consider the situation where at model development time t_a, we have access to a dataset of condition monitoring signals and system model estimates of process variables. Certainty about healthy operative conditions are only known until a past point in time t_b when an assessment of the system health was performed and declared healthy. Hence, at model development time, we only have access to the true healthy class for a portion of our data and fault signatures of an unknown number of fault types may be present in the remaining dataset. In particular, we consider the scenario where an evolving fault condition is actuality present but has not been detected due to the low intensity of the fault. In addition to the unlabelled data, an independent test set with increased levels of component degradation is provided. Our task is then to the detect the fault types in both, the unlabelled dataset and the test set. It should be noted that at t_a an incomplete knowledge of the world is present. Hence, we have an *open set* problem (Scheirer, de Rezende Rocha, Sapkota, & Boult, 2013) where we only know the initial healthy state but do not have any information on the faulty conditions. Therefore, not all possible classes are know at the model development stage and it is not even known how many fault classes may evolve. The

formulation of the diagnostic problem addressed in this paper is formally introduced in the following.

Given is a multivariate time-series of condition monitoring sensors readings from one unit $X_s = [x_s^{(1)}, \ldots, x_s^{(m)}]^T$, where each observation $x_s^{(i)} \in R^p$ is a vector of p raw measurements taken at operative conditions $w^{(i)} \in R^s$. In addition we have available residuals between measurements and estimated healthy system responses (i.e. $\delta x_s^{(i)}$) and the output of a calibrated system model that provides inferred values of the model tuners $\theta^{(i)}$ and estimates of the sensors readings $\hat{x}_s^{(i)}$ and virtual sensors $\hat{x}_v^{(i)}$. Hence, in compact form, we denote the complete set of measured and inferred inputs as $X = \{(w^{(i)}, \hat{x}_s^{(i)}, \hat{x}_v^{(i)}, \hat{\theta}^{(i)}, \delta_m^{(i)}))\}_{i=1}^m$. At model development time, the corresponding *true* system's health state is partially known and denoted as $H_s = [h_s^{(1)}, \ldots, h_s^{(m)}]^T$ with $h_s^{(i)} \in \{0,1\}$ where the healthy class is labeled as $h_s^{(i)} = 1$. Therefore, our partial knowledge of the true health allows to define two subsets of the available data: a *labeled* dataset $\mathcal{D}_L = \{(x^{(i)}, h_s^{(i)})\}_{i=1}^u$ with $h_s^{(i)} = 1$ corresponding to known healthy operative conditions and an *unlabeled sample* $\mathcal{D}_U = \{x^{(i)}\}_{i=u+1}^m$ with unavailable health labels. In particular, we consider scenarios where K unknown faults types are present in D_U. The fault types correspond to increasing intensities of the same fault mode (i.e. step-wise increases). The level of component degradation in D_U is low (i.e. $\leq -1\%$ nominal conditions) and therefore we represent the situation where faults signatures are present but are not yet detected at analysis time. In addition, we test the generalization capability of our model to detect K_* new faults of higher intensity in a test dataset $\mathcal{D}_\mathcal{T} = \{(x_*^{(j)}\}_{j=1}^M$. An schematic representation of the problem is provided in Figure 8.

Given this set-up we first consider the problem of detecting the faulty operative within $\{D_U, D_T\}$ given only our healthy dataset $\mathcal{D}_L$ at time t_a. Hence our initial task is to estimate the health state (i.e $\hat{\mathbf{h}}_\mathbf{s}$) on $\{D_U, D_T\}$. Furthermore, we aim to provide an isolation of the fault mode present. We refer to $\mathbf{V} = \{V_j | j = 1, \ldots, R\}$ as the partition of $\{D_U, \mathcal{D}_\mathcal{T}\}$ according to the $R = K + K_* + 1$ true but unknown fault types we aim to detect. For simplicity we will refer to the dataset $\{D_U, D_T\}$ as the combined test set that we denote as D_{T+}.

4.2. System Model Calibration

A conventional way to ensure that the system response follows observations X_s is to infer the values of the model correcting parameters θ solving an inverse problem. Since both the measurement data and model parameters are uncertain, the process of estimating optimal correcting parameters is a stochastic calibration problem. Ideally, the calibration process aims at obtaining the posterior distribution of the calibration factors given the data $p(\theta|w, X_s)$. However, computing the whole distribution is generally computationally very ex-
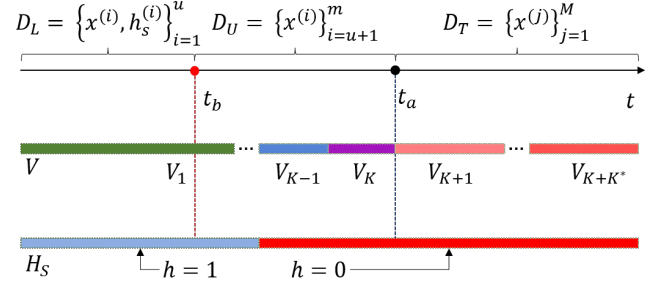

Figure 8. Schematic representation of the problem. Training dataset D has labelled (D_L) and unlabelled data (D_U). The test set (D_T) has only unlabelled samples. The true health condition any point in time is represented by the H_S bar. Healthy condition are represented in blue and faulty in red. The true operative condition type within the data are represented by the V bar. The healthy condition is shown in green; each fault type appear in a different color. K fault classes are present in D_U and K_* in D_T.

pensive and therefore in most cases, point value estimations of the parameters are inferred. A typical compromise is to resort to the mode of the posterior distribution that is called the *maximum a posteriori estimation* (MAP), described by

$$\hat{\theta}_{\text{MAP}} = \arg\max_\theta p(\theta|w, X_s) \qquad (2)$$

Several calibration methods have been proposed and the large majority of them can be classified as probabilistic matching approaches. Some of the most commonly used calibration methods include weighted linear and non-linear least squares schemes, maximum likelihood estimates or Bayesian inference methods (e.g. Markov Chain Monte Carlo, Particle and Kalman filters) (Arias Chao et al., 2015). These methods differ in the level of complexity and the computational cost.

In this work, we propose an Unscented Kalman Filter (Julier & Uhlmann, 1997) to infer the values of the model correcting parameters $\hat{\theta}$ since our models of interest are nonlinear. However, the task can be also performed with other approaches. Hence, rather than focusing on one particular model calibration method, we evaluate the impact of different levels of calibration accuracy in the performance of the proposed fault detection and isolation algorithm and therefore in the framework proposed.

Model-based estimation of the sub-model health parameters from a transient data stream can be addresses with a traditional state-space formulation. In particular we consider an UKF where the state vector comprises the health parameters. The measurement equation depends on the states and the input signals at the present time step t; which is readily available from the system model S. Hence, we apply a UKF to a

nonlinear discrete time system of the form:

$$\theta^{(t)} = \theta^{(t-1)} + \xi^{(t)} \tag{3}$$

$$x_s^{(t)} = S(w^{(t)}, \theta^{(t)}) + \epsilon^{(t)} \tag{4}$$

where $\xi \sim N(0, Q)$ is a Gaussian noise with covariance Q and $\epsilon \sim N(0, R)$ is a Gaussian noise with covariance R. A more detailed explanation of this problem formulation applied to the monitoring of gas turbine engines can be found in Borguet (Borguet, 2012).

The Kalman Filter provides estimates of $\hat{\theta}^{(t)}$ and therefore a fault detection and isolation (FDI) logic is also required. The standard approach for the FDI logic is to define thresholds for all $\hat{\theta}^{(t)}$ and rise an alarm if the inferred unobserved model parameter surpasses the defined threshold. However, such a approach has some limitations since a fault signature can be manifested only as a subtle change of the degradation rate. Accordingly, a threshold based approach will not be able to detect the presence of the fault signature until the fault is clearly manifested which will result in a detection delay. Similarly, a fault mode can result in different degradation rates in several of the monitored internal model parameters. As a result, fault signatures that are ambiguous when considered in each of the individual dimension of θ can be more clearly detected when considered combined. Therefore, we propose an alternative fault detection and isolation logic that is able to overcome some of the limitations of the threshold model-base approaches or a pure data-driven diagnostics models.

It should be noted that the proposed formulation of the calibration problem assumes that the system model has a good representation of the real physical process. This is a common situation when evaluating the health state of mature products where the system model has been developed and validated based on multiple field units or test beds units. In contrast, this is not the case of new developments. In the general, no model is perfect and a certain level of missing physical representation on the system model will imply in a lower calibration quality. In case of significant missing physics within the system model representation, the impact of model degradation gets entangled with the model correction rooted in the lack of physics. To mitigate this situation the calibration problem needs to be reformulated to account for a model discrepancy term $\delta(w)$ as follows:

$$X_s = S(w, \theta(w)) + \delta(w) + \epsilon \tag{5}$$

$$\theta(w) = f(w) + \xi \tag{6}$$

Hence, the solution of this reformulated calibration problem involves finding the functions $\delta(w)$ and $\theta(w)$. The current state of the art solution is a sequential process. The equation is first solved for $\theta(w)$ and subsequently for $\delta(w)$. However,

this approach is not optimal since smearing between the correction in θ and $\delta(w)$ is typically present in the solution. On the other hand, the simultaneous solution of $\theta(w)$ and $\delta(w)$ is an open research question that we do not address in this paper.

4.3. Fault Detection

Several approaches for fault detection problems have been proposed in the literature. One of the main distinction criteria between them is the availability of labeled data. If labeled data from faulty and healthy operation are available, the problem is typically defined as a binary classification. However, faulty system conditions in critical systems are rare resulting in relatively few or even no faulty condition monitoring data. The focus of this paper is on the latter scenario, for which we define the problem as a *one-class classification* (Moya & Hush, 1996).

One-Class Classification. The fault detection problem has been successfully addressed as a *one-class classification* problem in (Michau, Hu, Palmé, & Fink, 2017). In this case the task turns to a regression problem that aims to discover a functional map $\mathcal{G}$ from the healthy operation conditions to a target label $\mathbf{T} = \{h^{(i)} \mid x^{(i)} \in S_T\}$ where $S_T \subsetneq \mathcal{D}_L$ is a training subset of $\mathcal{D}_L$. We consider a neural network model to discover the functional map $\mathcal{G}$ and hence we refer to such a network as the *one-class network*. The output of the one-class network will deviate from the target value $\mathbf{T}$ when the inner relationship of a new data point $x^{(j)} \in \mathcal{D}_{T+}$ does not correspond to the one observed in S_T. Therefore, we consider an unbounded similarity score $s_I(x^{(j)}; \beta)$ of $x^{(j)}$ with respect to our healthy labeled data based on the absolute error of the prediction $\mathcal{G}(x^{(j)})$ that we define as follows:

$$s_I(x^{(j)}; \beta) = \frac{\mid \mathbf{T} - \mathcal{G}(x^{(j)}) \mid}{\beta} \tag{7}$$

$$\beta = P_{99.9}(\mid (\mathbf{T} - \mathcal{G}(S_V) \mid)\gamma \tag{8}$$

where β corresponds to a normalizing threshold given by the 99.9% percentile of the absolute error of the prediction of $\mathcal{G}$ in a validation set (i.e S_V) extracted from $\mathcal{D}_L$ multiplied by a safety margin $\gamma = 1.5$. Please note, that the percentile and γ are hyper-parameters and can be adjusted to the specific problem.

Hence, our fault detection algorithm is simply given by:

$$\hat{h}_s(x^{(j)}) = \begin{cases} 1 & s_I(x^{(j)}; \beta) < 1 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

To obtain the mapping function $\mathcal{G}$ we resort to a partially supervised learning strategy with embedding given only one target label $h_s^{(i)} = 1$ at training.

Partially Supervised One-Class Learning with Embedding. The goal of a supervised learning strategy to discover a direct mapping from input X to a target label T given a training set S_T. An alternative strategy to this direct mapping is to obtain a representation of the raw input data (a.k.a. non-linear embedding) from which a reliable optimal mapping $\mathcal{G}$ can be learned. Hence, the task has two parts. Firstly, we find a transformation $E : X_L \longmapsto \mathbf{z_L}$ of the input signals to a latent space $\mathbf{z_L} \in R^{u \times d}$ that encode optimal distinctive features of X_L in an unsupervised way (i.e. without having information on the labels). In a second step, we find a functional mapping $\mathcal{G}_{sle} : \mathbf{z_L} \longmapsto \mathbf{T}$ from the latent representation of input signals $\mathbf{z_L} = \{E(x^{(i)}) \mid x^{(i)} \in S_T\}$ to the label class $\mathbf{T}$. Since in our one-class problem formulation the training target contains only one class and since the number and nature of the fault classes in $\mathcal{D}_{T+}$ are not known in advance, we denote the corresponding supervised problem as partially supervised learning. This is a key difference to conventional supervised learning diagnostics where the available labeled (training) data samples already cover the essential information on the number and type of classes and the new observation only fall in the category of already known classes.

Different unsupervised deep-learning models can be considered to discover the latent representation z_L. In order to cover the most prominent deep neural network architectures and to show the performance independence of our proposed hybrid method to the network architectures we implemented two discriminative and one generative autoencoder variants. For the discriminative autonencodes, we considered vanilla autonecoders (AE) and hierarchical extreme learning machines (HELM) (Zhu, Miao, Qing, & Huang, 2015). For the generative methods we implemented variational autoencoders (VAE) (Kingma & Welling, 2014). For the one-class network we use a discriminative model based on a feed-forward network (FF). A formal introduction to the selected neural networks model is provided in Section 9.

It should be noted that our proposal for an embedding representation is not related to the quality of the one-class network to discriminate healthy and faulty conditions but to the need of performing fault isolation. The detection problem can also be formulated without an embedding (i.e. direct mapping from input X to a target label T given a training set S_T).

4.4. Fault isolation

The autoencoder formulation of the problem allows to compute the expected signal values under the training distribution (i.e. X). The output of the autoencoder network $F(x^{(j)})$ will deviate from the input value X when the inner relationship of a new data point $x^{(j)} \in \{\mathcal{D}_U, \mathcal{D}_\mathcal{T}\}$ does not correspond to the one observed in the training set S_T. Therefore, we compute the absolute deviation that each component of the reconstructed signals has (i.e. $|x_k^{(j)} - F(x^{(j)})_k|$) relative to the error observed in the validation dataset S_V (i.e. healthy operation conditions).

$$d_I(x_k^{(j)}; \nu_k) = \frac{|x_k^{(j)} - F(x^{(j)})_k|}{\nu_k} \tag{10}$$

where ν corresponds to a normalizing threshold given by the 99.9% percentile of the absolute error of the prediction of F in the validation set S_V

$$\nu_k = P_{99.9}(\{|x_k^{(j)} - F(x^{(i)})_k| \mid x_k^{(i)} \in S_V\}) \tag{11}$$

$d_I(x_k^{(j)}; \nu)$ is an unbounded measure of similarity between the signal value predicted by the autoencoder network and the expected or true signal value. In our hybrid approach, the input space to the autoencoder comprises the calibration factors θ and the observed signals X_s and therefore deviations in the signal reconstruction can be pointed out for measurement and model tuning factors.

5. CASE STUDY

5.1. A Single Fault Mode in a Turbofan Engine

A new dataset was designed to evaluate the proposed methodology. The CMAPSS dataset DS00 provides simulated condition monitoring data of an advanced gas turbine during 24 flights cycles. The dataset was synthetically generated with the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dynamical model (Frederick et al., 2007). Real flight conditions as recorded on board of a commercial jet were taken as input to the C-MAPSS model (*DASHlink - Flight Data For Tail 687*, 2012). Figure 9 shows 14 simulated flight envelopes given by the traces of altitude (alt), flight Mach number (XM), throttle-resolver angle (TRA) and total temperate at the fan inlet (T2). Each flight cycle contains $\sim$175 snapshots of recordings covering climb, cruise and descend flight conditions (i.e. alt $>$ 10000 ft). The labeled dataset $\mathcal{D}_L$ (blue) consists of 20 flight cycles with a healthy state of the engine (i.e $h_s = 1$). The unlabeled and test datasets $\{\mathcal{D}_U, \mathcal{D}_\mathcal{T}\}$ (green and red respectively) contain snapshots of $R = 4$ concatenated flight cycles with a deteriorating engine condition. The intensity of the degradation increases at each flight (i.e step-wise increase). The fault mode corresponds to a high pressure compressor (HPC) efficiency degradation. Each of the fault magnitudes is denoted with a fault id (see Table 1). The unlabeled dataset also includes 60 snapshots of initial healthy operation. The unlabeled and test datasets $\{\mathcal{D}_U, \mathcal{D}_\mathcal{T}\}$ contain a subset of flight conditions experienced during training. Therefore, this set-up relates to a real scenario where an aircraft is operating under certain flight routes, which results in very similar flight condition. In addition to the noisy flight conditions, all the healthy operative conditions incorporate white noise in all the engine health model parameters (see Table 5). No additional noise or bias is considered for sensor readings. A total of $\sim$ 3200 healthy

data points are available for training. The unlabeled and test datasets $\{\mathcal{D}_U, \mathcal{D}_\mathcal{T}\}$ contain ~ 740 data points.
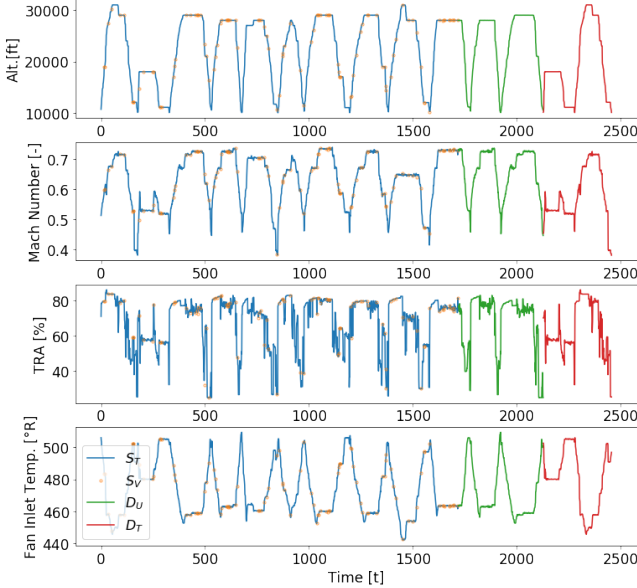

Figure 9. Subset of 10 flight envelopes given by the traces of altitude (top), flight Mach number (middle) and throttle-resolver angle -TRA (bottom). Four dataset are shown: S_T (blue), S_V (orange), D_U (green) and D_T (red).

Fault Id	Fault Mode	Magnitude	dataset
1	HPC Efficiency	-0.5 %	$\mathcal{D}_U$
2	HPC Efficiency	-1.0 %	$\mathcal{D}_U$
3	HPC Efficiency	-1.5 %	$\mathcal{D}_T$
4	HPC Efficiency	-2.0 %	$\mathcal{D}_T$

Table 1. Overview of the generated faults

5.2. Pre-processing

The dimension of the input space X (i.e. n) varies depending on the solution strategy considered (see Table 2). The diagnostic model based on condition monitoring data (CMBD) has 17 inputs. The residual-based model considers 31 inputs and the proposed hybrid method based on system model calibration (CBHD) uses to 45 inputs. Tables 3 to 6 provide a detailed overview of the model variables included in the condition monitoring signals $[W, X_s]$, model residuals δ_{X_s}, virtual sensors $\hat{X}_v$ and model calibration parameters $\hat{\theta}$.

Model	Input	n
CMBD - No Hybrid	$[W, X_s]$	17
Residual	$[W, \delta_{X_s}]$	31
CBHD - Hybrid	$[W, \hat{X}_s, \hat{X}_v, \hat{\theta}]$	45

Table 2. Dimension of the input space for the autoencoder network - n

Id	Symbol	Description	Units
1	alt	Altitude	ft
2	XM	Flight Mach number	-
3	TRA	Throttle-resolver angle	%
4	Wf	Fuel flow	pps
5	Nf	Physical fan speed	rpm
6	Nc	Physical core speed	rpm
7	T2	Total temperature at fan inlet	$^\circ$R
8	T24	Total temperature at LPC outlet	$^\circ$R
9	T30	Total temperature at HPC outlet	$^\circ$R
10	T48	Total temperature at HPT outlet	$^\circ$R
11	T50	Total temperature at LPT outlet	$^\circ$R
12	P15	Total pressure in bypass-duct	psia
13	P21	Total pressure at fan outlet	psia
14	P24	Total pressure at LPC outlet	psia
15	Ps30	Static pressure at HPC outlet	psia
16	P40	Total pressure at burner outlet	psia
17	P50	Total pressure at LPT outlet	psia

Table 3. Condition monitoring signals - $[W, X_s]$. The Id is used in this document as shorthand of the variable description. The variable symbol corresponds to the internal variable name in CMAPSS. The descriptions and units are reported as in the model documentation (Frederick et al., 2007).

The input space X to the models is normalized to a range $[-1, 1]$ by a min/max-normalization. A validation set $S_T \subsetneq D_L$ comprising 6 % of the labelled healthy data for all the models was chosen.

5.3. Network architectures

The partially supervised with embedding learning strategies require an *autoencoder network* in addition to the *one-class network*. As shown in Figure 10, the input signals X are reconstructed by the encoder-decoder networks. The encoder provides a new representation z of the input signals. The mapping to the target label $\mathbf{T}$ is carried out by the *one-class network* taking as input the latent (i.e. unobserved) representation of the input data z.

To evaluate the different methods in a fair way, we separate the effect of regularization in the form of model and learning strategies choice from other inductive bias in the form of choice of neural network architecture. Therefore, we define a common architecture of the *one-class network* and *autoencoder network* for all our deep autoencoders.

One-Class Network. The proposed network topology uses three fully connected layers ($L = 3$). The first hidden layer has 20 neurons (i.e. $m^1 = 20$), the last hidden layer has 100 neurons ($m^{L-1} = 100$). The network ends with a linear output neuron ($m^L = 1$). Therefore, in compact notation, we refer to the *one-class network* architecture as $[20, 100, 1]$. *tanh* activation function is used throughout the network. It should be noted that the one-class classification problem formula-

Id	Symbol	Description	Units
18	T40	Total temp. at burner outlet	°R
19	P30	Total pressure at HPC outlet	psia
20	P45	Total pressure at HPT outlet	psia
21	W21	Fan flow	pps
22	W22	Flow out of LPC	lbm/s
23	W25	Flow into HPC	lbm/s
24	W31	HPT coolant bleed	lbm/s
25	W32	HPT coolant bleed	lbm/s
26	W48	Flow out of HPT	lbm/s
27	W50	Flow out of LPT	lbm/s
28	epr	Engine pressure ratio (P50/P2)	–
29	SmFan	Fan stall margin	–
30	SmLPC	LPC stall margin	–
31	SmHPC	HPC stall margin	–
32	NRf	Corrected fan speed	rpm
33	NRc	Corrected core speed	rpm
34	PCNfR	Percent corrected fan speed	pct
35	phi	Ratio of fuel flow to Ps30	pps/psi

Table 4. Virtual sensors - $[X_v]$. The Id is used in this document as shorthand of the variable description. The variable symbol corresponds to the internal variable name in CMAPSS. The descriptions and units are reported as in the model documentation (Frederick et al., 2007).

Id	Symbol	Description	Units
36	fan_eff_mod	Fan efficiency modifier	-
37	fan_flow_mod	Fan flow modifier	-
38	LPC_eff_mod	LPC efficiency modifier	-
39	LPC_flow_mod	LPC flow modifier	-
40	HPC_eff_mod	HPC efficiency modifier	-
41	HPC_flow_mod	HPC flow modifier	-
42	HPT_eff_mod	HPT efficiency modifier	-
43	HPT_flow_mod	HPT flow modifier	-
44	LPT_eff_mod	LPT efficiency modifier	-
45	LPT_flow_mod	HPT flow modifier	-

Table 5. Model correcting parameters - $[\theta]$. The Id is used in this document as shorthand of the variable description. The variable symbol corresponds to the internal variable name in CMAPSS. The descriptions and units are reported as in the model documentation (Frederick et al., 2007).

tion is a regression problem and therefore the last activation $\sigma^L = I$ is the identity.

Autoencoder Networks. Based on the same argument as mentioned above, the autoencoder models (i.e. AE and VAE) use the same encoder architecture with two hidden layers ($l_z = 2$) with $m^1 = 20$ and latent space of 8 neurons ($d = 8$). In compact notation, we refer to the *autoencoder network* architecture as $[n, 20, 8, 20, n]$. Where n denotes the size if the input space X; which varies depending on the solution strategy considered. The VAE model uses the mean of the approximate posterior (i.e. μ) as the model latent space to avoid using approximate samples from posterior distribution (i.e. $z^{(i)}$). The HELM model reproduces the encoder and the one-class networks in one single hierarchical network. Hence, the resulting network architecture is $[n, 20, 8, 20, 100, 1]$.

Id	Symbol	Description	Units
δ_4	δ_{Wf}	Delta fuel flow	pps
δ_5	δ_{Nf}	Delta physical fan speed	rpm
δ_6	δ_{Nc}	Delta physical core speed	rpm
δ_7	δ_{T2}	Delta total temp. at fan inlet	°R
δ_8	δ_{T24}	Delta total temp. at LPC outlet	°R
δ_9	δ_{T30}	Delta total temp. at HPC outlet	°R
δ_{10}	δ_{T48}	Delta total temp. at HPT outlet	°R
δ_{11}	δ_{T50}	Delta total temp. at LPT outlet	°R
δ_{12}	δ_{P15}	Delta total press. in bypass-duct	psia
δ_{13}	δ_{P20}	Delta total press. at fan inlet	psia
δ_{14}	δ_{P24}	Delta total press. at LPC outlet	psia
δ_{15}	δ_{Ps30}	Delta static press. at HPC outlet	psia
δ_{16}	δ_{P40}	Delta total press. at burner outlet	psia
δ_{17}	δ_{P50}	Delta total press. at LPT outlet	psia

Table 6. Delta to healthy state - $[\delta_{X_s}]$. The Id is used in this document as shorthand of the variable description. The variable symbol corresponds to the internal variable name in CMAPSS. The descriptions and units are reported as in the model documentation (Frederick et al., 2007).

OC-SVM model. To evaluate if a deep learning architecture is required for the fault detection task, we compare the results to the standard one-class support vector machines (One-Class SVM) (Schölkopf et al., 2000) for novelty detection. This enables us to evaluate the benefits and the potential need for complex neural network architectures for the defined fault detection task. We use the standard *scikit-learn* (Pedregosa et al., 2011) implementation of the one class SVM with an radial basis function kernel, *nu=0.001* and *gamma=0.1*. The model performance is sensitive to the choice of these hyper-parameters. Therefore, optimal parameters for the validation set S_V may not guarantee a good performance in D_T. We selected the hyperparameters that maximize the F_1 score on the test set to ensure the best possible performance of the baseline on the test dataset. For other algorithms, the parameters were selected based on the validation dataset. This makes the comparison of the deep learning algorithms to the baseline even more challenging.

5.4. Training Set-up

The optimization of the networks' weights of all the models was carried out with mini-batch stochastic gradient descent (SGD) and with the *Adam* algorithm (Kingma & Ba, 2015). *Xavier* initializer (Glorot & Bengio, 2010) was used for the weight initializations. The learning rate (LR), epoch and batch size were set according to Table 7. The batch size for the autonecoder network was set to 512 and to 16 for one-class network. Similarly, the number of epochs for autoencoder training was set to 2000 and for the supervised models to 500. Therefore, all these methods use the same network architecture and hyper-parameters for the optimisation.
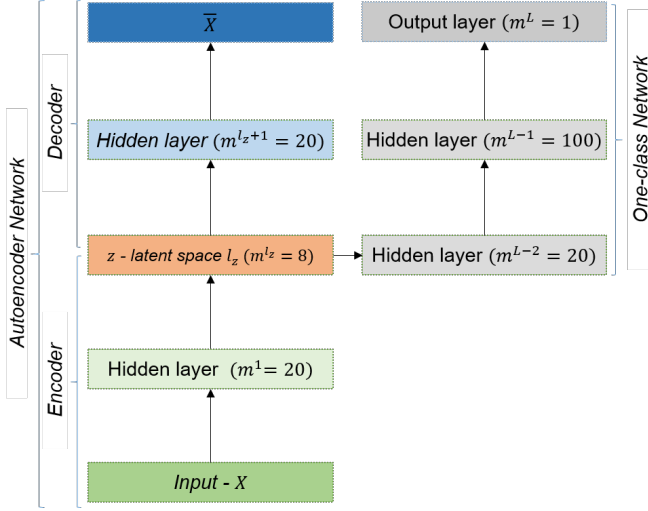
Figure 10. Network architecture for the defined learning problem with an autoencoder (encoder-decoder) and the one-class detection network.

Model	LR	Batch Size	Epochs
One-class	0.001	16	500
Autoencoder	0.001	512	2000

Table 7. Training parameters

5.5. Evaluation Metrics

In order to compare and analyse the performance of our models on the intended diagnostics task we defined two evaluation aspects: detection of unknown faults (i.e. estimation of h_s) and fault isolation. For each of the two aspects, we consider targeted evaluation metrics that are defined in the following.

Fault Detection. Given the combined test dataset D_{T+} with true health state $h_s^{(j)}$ and the corresponding estimated health state $\hat{h}_s^{(j)}$, we evaluate the performance of the fault detection algorithm as the accuracy of a binary classification problem

$$\text{Acc} = \frac{1}{M+m} \sum_{i=1}^{M+m} \mathbf{1}(h_s^{(j)} = \hat{h}_s^{(j)}) \qquad (12)$$

where $M+m$ number of data points in D_{T+} and $\mathbf{1}\{.\}$ denotes the indicator function.

Fault Isolation. The error in the reconstruction signal will be more notorious for those signals in close relation to the fault root cause. Therefore, we report the index of the signals of those components of the data point $x^{(j)}$ that satisfy $d_I(x_k^{(j)}; \nu_k) > 1$. The mapping between variable description of the variables and the corresponding index is provided in Tables 3 to 6.

6. EXPERIMENTAL RESULTS

6.1. Fault Detection

Table 8 shows the performance of our twelve models on fault detection. The residual-based and the calibration-based approaches achieve nearly 100% detection accuracy independently of the neural network model considered. Both approaches provide an improvement of nearly 80% with respect to the best diagnostic model based purely on condition monitoring data (i.e. *AE* with $[W, X_s]$). The OC-SVM model results in a lower detection accuracy than the deep learning models with independence on the input space considered. Concretely, the best performing autoencoder model (i.e. AE) provides c.a. 4% accuracy improvement with respect to the standard OC-SVM when a calibration-based approaches is considered.

Input	AE	VAE	HELM	OC-SVM
$[W, X_s]$	24.5	12.9	8.0	10.0
$[W, \delta_{X_s}]$	98.7	99.3	8.5	79.5
$[W, \hat{X}_s, \hat{X}_v, \hat{\theta}]$	**100.0**	99.1	97.5	96.2

Table 8. Overview of detection results - Accuracy in [%]. Mean values of 10 runs

Diagnostics models based on condition monitoring data show poor performance independently on the autoencoder network considered. A possible explanation for this may be that this is due to the high complexity of the dataset in the form of a large variability in the input space due to varying operating conditions. To verify this idea we trained the *AE* model with CM inputs on a subset of the training data with operative points closer to cruise conditions. Hence, we restricted the fight altitude to above 25000 ft. Figure 11 shows the accuracy of the diagnostics model based on condition monitoring data trained in this simpler dataset. We can observe that the detection performance drastically increases; which supports our hypothesis.

The detection performance of the one-class solutions reported in Table 8 is determined by the capability of the similarity score $s_I(x^{(i)}; \beta)$ to represent a valid and consistent distance to healthy operation learnt in the training phase (i.e. D_L). To demonstrate and verify this behaviour we plot in Figure 12 the similarity score obtained with *AE* model with CM inputs in the four HPC efficiency faults of increasing intensities (-0.5% to -2%). The onset of each fault is indicated by the dashed vertical lines. We observe that the more severe the fault is, the higher the detection index. Therefore, $s(x^{(i)}; \beta)$ shows the expected consistency. However, we also observe that only for HPC faults with intensities below -1.0% the similarity score is above the decision threshold $s(x^{(i)}; \beta) > 1$ (black horizontal line). Hence, the one-class network fails to discriminate between healthy and faulty conditions for HPC efficiency deterioration below 1.0%.
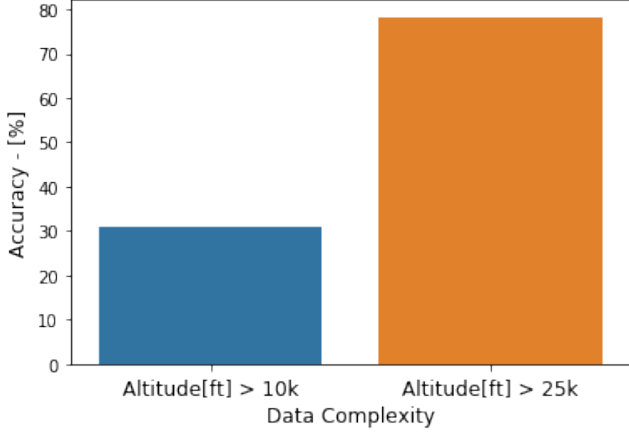
11

Figure 11. Evolution of the accuracy with dataset complexity for AE model based on $[W, X_s]$ inputs in faults 1, 2 and 4. Fault 3 is not considered since $alt < 25000$ ft.

Figure 12. Similarity index for four HPC efficiency faults of different intensities with AE model based on $[W, \hat{X}_s]$ signals. All the faults occur at different flight conditions. The decision threshold is plotted as horizontal black line ($s = 1$). The onset times of each fault are indicated by the vertical dashed lines. Four dataset are shown: S_T (blue), S_V (orange), D_U (green) and D_T (red).

The quality of the calibration process has an impact on the fault detection performance. In order to quantify this impact, the calibration factors are contaminated with noise of different signal-to-noise ratios. We impose the noise perturbation to all the calibration factors (i.e $\theta \in R^{10}$), however the impact is more pronounced for *HPC_Eff_mod* since it defines the fault mode. Figure 14 shows two components of the resulting noisy calibration process. Figure 13 shows the impact in fault detection accuracy of the different noise levels for two best performing models. We can observe a decrease in the accuracy as the noise increases for all the tested models. OC-SVM model shows the most robust performance for noise levels $SNR_{db} < 30$. Most of the models are able to achieve an accuracy that is above the pure data-driven models if all the SNR are evaluated. Therefore, these results demonstrate the robustness of the proposed fault detection approach. It should be noted that the SNR_{db} scale is logarithmic.

6.2. Fault Isolation

Table 9 shows the input signals detected as anomalous with the AE and VAE models. For simplicity, we report the index of the signals according to Tables 3 to 6. The affected signals are presented in a decreasing order according the value of the similarity indicator $d_I(x_k^{(j)}; \nu_k)$. Hence, the most affected signals are presented first. Only variables that satisfy $d_I(x_k^{(j)}; \nu_k) > 1$ are reported.

The four faults present in the combined test sets $\mathcal{D}_{T+}$ are rooted in a HPC efficiency deficit. However, not all the models have an input space where the compressor efficiency is represented. Concretely, only the calibration-based hybrid model with inputs $[W, \hat{X}_s, \hat{X}_v, \hat{\theta}]$ has a representation of the HPC efficiency through the estimated model correcting parameters $\hat{\theta}$. Therefore, in the best case, the remaining models can only aim to place the root cause of a HPC degra-
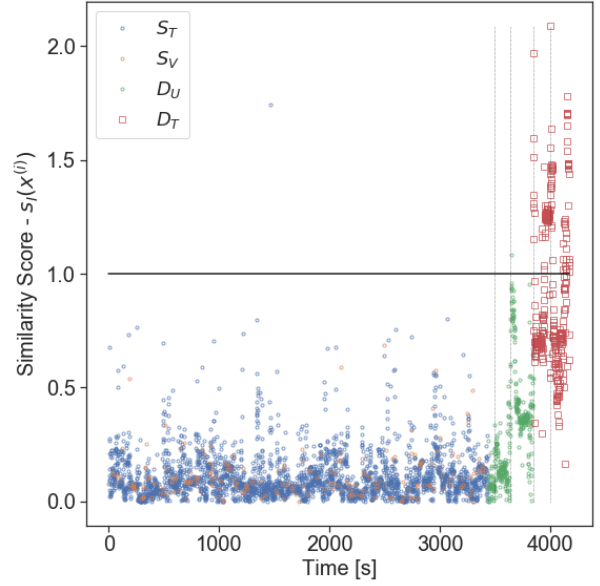
dation on variables physically related to the HPC. For instance, models that consider only condition monitoring signals $[W, X_s]$ detect a large reconstruction error in variable 6 (i.e. the rotational core speed of the shaft where the high pressure compressor is placed). The hybrid model based on residual $[W, \delta_{X_s}]$ encodes the fault signature in five residuals: δ_{11}, δ_{10}, δ_9, δ_6 and δ_8. Therefore, the residual of core speed δ_6 is also detected as an affected signal in addition to the HPC outlet temperature (δ_9) and temperatures at the outlets of the High and Low Pressure Turbines (i.e. δ_{10} and δ_{11}). The isolation of these last two process variables as the fault root cause is a clear smearing of the effect of an HPC degradation to other unrelated subsystems. Neural networks based on VAE show a similar isolation performance.

Finally, hybrid models based on calibrated models with input signals $[W, \hat{X}_s, \hat{X}_v, \hat{\theta}]$ encode the fault signature in only variable 40; which corresponds to the component of θ representing the correction of the HPC efficiency. Any model with $[W, \hat{X}_s, \hat{X}_v, \hat{\theta}]$ provides perfect isolation.

6.3. Feature Representation

The results presented in the previous section have demonstrated that the proposed hybrid approach provides a very good performance for fault detection and isolation, particularly for systems with a high variability of the operating conditions. To better understand how the different (expanded)
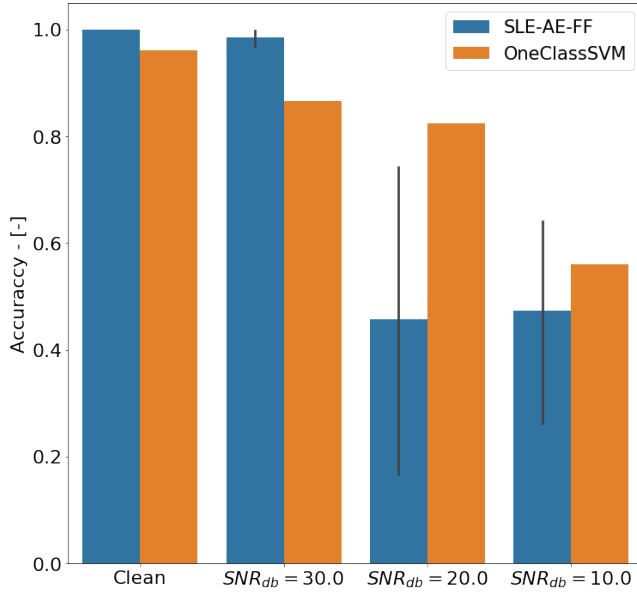
Figure 13. Fault detection accuracy as function of the noise levels for AE and OC-SVM model. 95% confidence intervals are shown as rectangular bars.

AE				
Input	-0.5%	-1.0%	-1.0%	-2.0%
$[W, X_s]$		-	6	6, 15
$[W, \delta_{X_s}]$	$\delta_{11}, \delta_{10}, \delta_9, \delta_6, \delta_8$			
$[W, \hat{X}_s, \hat{X}_v, \hat{\theta}]$	40			
VAE				
Input	-0.5%	-1.0%	-1.0%	-2.0%
$[W, X_s]$	-			
$[W, \delta_{X_s}]$	$\delta_{10}, \delta_{11}, \delta_6, \delta_9, \delta_8$			
$[W, \hat{X}_s, \hat{X}_v, \hat{\theta}]$	40			

Table 9. Overview of isolation results on four HPC efficiency faults with impact from -0.5% to -2.0%. The table shows the index of the affected variables as introduced in Tables 4-6. Variables affected by smearing are colored in red.

input spaces affect the latent representation and also the performance of the models on the diagnostics tasks, the latent space of the different models is analyzed. Please note that the analysis of the latent space is mainly performed for understanding and demonstration purposes. Therefore, only the first two dimensions of the latent space are visualized. While this does not provide a full evaluation of the latent space, a separability of the healthy and faulty conditions in the first two dimensions of the latent space would support the assumption that such a representation would also be favorable for the diagnostics tasks based on this latent representation.

Figure 15 shows a pairwise scatter plot of the first two dimensions of the latent space z of the hybrid AE model $X = [W, \hat{X}_s, \hat{X}_v, \hat{\theta}]$, while Figure 16 represents the first two dimensions of the latent space of the data-driven model $X =$
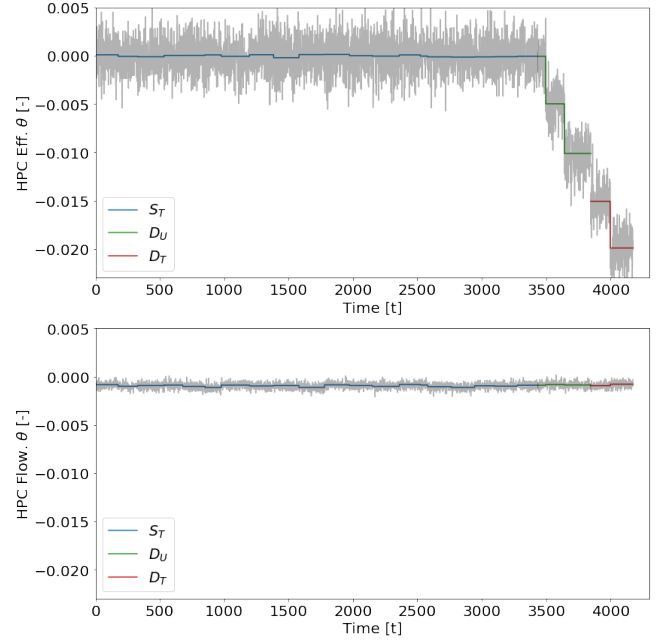

Figure 14. Noisy calibration factors for a noise level of $\text{SNR}_{db} = 10$ imposed on the high pressure compressor (HPC) efficiency. The added noise to the high pressure compressor (HPC) flow (which is not affected by the fault mode) is shown as reference. Three datasets are shown: S_T (blue), D_U (green) and D_T (red).

$[W, X_s]$. The latent space of the residual-based approach ($[W, \delta_{X_s}]$) is represented in Figure 17. The scatter plots are colored according to the dataset of the origin. The healthy class (i.e S_T) is shown in blue, healthy unlabelled data from $D_U(h_s = 1)$ are shown in orange and the faulty operative conditions from the test set D_T are represented in green.

It can be clearly observed that expanding the input space with additional model variables has a large impact in the latent representation. Concretely, the faulty conditions are clearly clustered together and have a high distance to the healthy operating conditions (centered around zero) for the two hybrid approaches. On the contrary, a distinction between healthy and faulty conditions in the latent representation of the purely data-driven approach $X = [W, X_s]$ is not possible. The representation of healthy and unhealthy classes shows clear overlaps in the two represented dimensions. These exemplary plots support the argument that the hybrid approaches provide a more favorable and more distinct representation of the healthy respectively unhealthy conditions. This results in a easier detection task of the one-class network leading to better detection results.

7. DISCUSSION

The performed experiments on the C-MAPSS dataset demonstrate that the proposed hybrid deep learning-based diagnostics algorithm, combining information from a physics-based
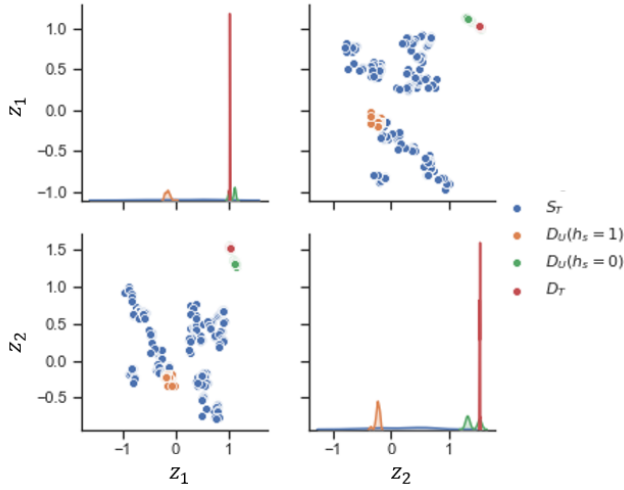
Figure 15. Pairwise scatter plot the first two components of the latent space z of the hybrid AE model with $X = [W, \hat{\theta}, \hat{X}_s, \hat{X}_v]$. The scatter plot is colored according to the dataset of origin: S_T (blue), $D_U(h_s = 1)$ (orange), $D_U(h_s = 0)$ (green) and D_T (red)
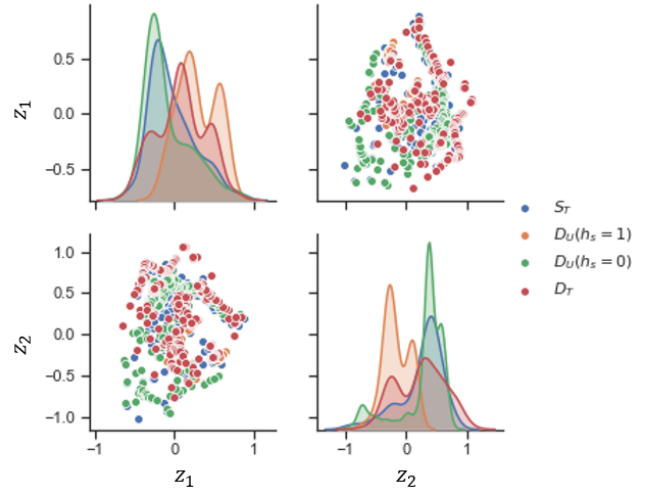

Figure 16. Pairwise scatter plot the first two components of the latent space z of the data-driven AE with $X = [W, X_s]$. The scatter plot is colored according to the dataset of origin: S_T (blue), $D_U(h_s = 1)$ (orange), $D_U(h_s = 0)$ (green) and D_T (red)

model and condition monitoring data, outperforms pure data-driven deep learning methods in fault detection and isolation, particularly for systems with a high variability of operating conditions.

Standard residual-based approaches (i.e. using δ_{X_s}) and hybrid approaches based on calibrated model variables (i.e using $[\hat{X}_s, \hat{X}_v, \hat{\theta}]$) led to similar fault detection performance. The analysis of the encoded representations showed that their excellent detection performance is rooted in the same concept. Both latent spaces provide a clear discrimination between healthy and faulty operating conditions; which simplifies the fault detection task. This result implies that an accurate model calibration is not relevant to obtain good detection performance as long as the system degradation or fault signature is encoded in model inferred variables (i.e. $\hat{\theta}$, δ_{X_s} or both).

However, accurate fault isolation (overcoming the smearing effect) is only possible when model tuning parameters $\hat{\theta}$ are considered. Hence, the proposed hybrid approach based on calibrated inputs provides clear benefits for the fault isolation task. However, it should be noted that this approach introduces an additional pre-processing step. Also, the performance of this approach depends on the calibration capabilities and it is expected that if the calibrated model fails to reproduce closely the reality, the capability to clearly isolate failures will decrease.

Residual based and calibration based frameworks are not mutually exclusive and therefore a third option is to combine them. In this case, in addition to a pre-processing calibration step, the residuals δ_{X_s} to a healthy system state are also computed. Hence, the input to the deep-learning diagnostics model would comprise $[w, \hat{X}_s, \hat{X}_v, \hat{\theta}, \delta_{X_s}]$.

8. CONCLUSIONS

In this paper, we proposed a hybrid fault diagnosis framework combing the physical performance models with deep learning algorithms.

The performance of the proposed framework was evaluated on a synthetic dataset generated with the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dynamical model. The C-MAPSS dataset D00 provides simulated condition monitoring data of an advanced gas turbine during real flight conditions under healthy and four faulty operative condition.

The proposed framework and method was able to outperform purely data-driven deep learning algorithms and the traditional OC-SVM model for fault detection (providing a perfect detection accuracy) and for fault isolation (being able to precisely isolate the root cause of the originating fault). The proposed methodology is able to overcome the smearing that is commonly observed in purely data-driven approaches where all the affected signals and not the root cause are isolated by the algorithms.

More importantly, we showed that the advantages of hybrid models are particularly relevant for complex datasets with a large variability in the operating conditions. Under these conditions, purely data-driven deep-learning approaches derived from condition monitoring data fail to obtain a robust diagnostic model. However, for systems with more homoge-
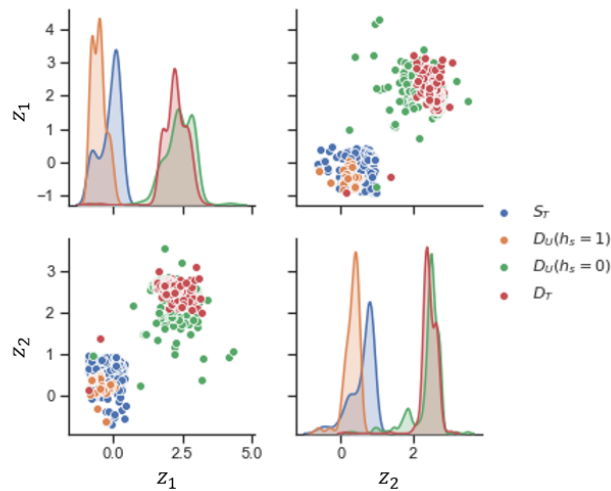
Figure 17. Pairwise scatter plot the first two components of the latent space z of the hybrid delta AE model with $X = [W, \delta_{X_s}]$. The scatter plot is colored according to the dataset of origin: S_T (blue), $D_U(h_s = 1)$ (orange), $D_U(h_s = 0)$ (green) and D_T (red)

neous operating conditions, we expect a similar performance between the hybrid and the data-driven approaches for fault detection tasks.

A feature learning analysis indicates that the excellent detection results obtained with hybrid methods are rooted in the fact that the latent space z provides a representation of the input signals that is clearly informative about the true label class.

As demonstrated in the experiments, accurate isolation results are obtained when the calibrated system model has a good representation of the fault modes. However, the analysis of fault modes that are not represented in the system model is of interest for practical applications. In this situation, it could be expected that the calibrated model fails to reproduce closely the reality and the capability to isolate faults will decrease. The situation can be mitigated by considering residuals between measurements and the estimated model responses or incorporating these residuals in the calibration process. The analysis of these possible scenarios and the verification of the real potential of the proposed solution in a more realistic setting is subject of further research.

REFERENCES

Arias Chao, M., Lilley, D. S., Mathé, P., & Schloßhauer, V. (2015). Calibration and Uncertainty Quantification of Gas Turbine Performance Models. In *Proceedings of the asme turbo expo* (Vol. 7A, p. V07AT29A001). doi: 10.1115/gt2015-42392

Baraldi, P., Di Maio, F., Turati, P., & Zio, E. (2015). Robust signal reconstruction for condition monitoring of industrial components via a modified auto associative kernel regression method. *Mechanical Systems and Signal Processing*, *60*, 29–44.

Borguet, S. (2012). *Variations on the Kalman Filter for Enhanced Performance Monitoring of Gas Turbine Engines* (PhD Thesis). Université de Liège.

Brunell, B. J., Mathews, J. H. K., & Aditya Kumar. (2004). *United States Patent Design of an Adaptive Model-Based Control for Controlling a Gas Turbine* (Vol. 121).

DASHlink - Flight Data For Tail 687. (2012). Retrieved 2019-01-29, from https://c3.nasa.gov/dashlink/

Doersch, C. (2016, jun). *Tutorial on Variational Autoencoders* (Tech. Rep.). Retrieved from http://arxiv.org/abs/1606.05908

Dourado, A., & Viana, F. A. C. (2019, sep). Physics-Informed Neural Networks for Corrosion-Fatigue Prognosis. *Proceedings of the Annual Conference of the PHM Society*, *11*(1). doi: 10.36001/PHM-CONF.2019.V11I1.814

Frank, S., Heaney, M., Jin, X., Robertson, J., Cheung, H., Elmore, R., & Henze, G. P. (2016). Hybrid Model-Based and Data-Driven Fault Detection and Diagnostics for Commercial Buildings. *Proceedings of the ACEEE Summer Study on Energy Efficiency in Buildings, Aug 21-26*, 12.1–12.14.

Frederick, D. K., Decastro, J. A., & Litt, J. S. (2007). *User's Guide for the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS)* (Tech. Rep.). Retrieved from http://www.sti.nasa.gov

Glorot, X., & Bengio, Y. (2010). *Understanding the difficulty of training deep feedforward neural networks* (Tech. Rep.). Retrieved from http://www.iro.umontreal.

Hanachi, H., Yu, W., Kim, I. Y., & Mechefske, C. K. (2017). Hybrid Physics-Based and Data-Driven PHM. *Canadian Machinery Vibration Association (CMVA) Annual Conference, Edmonton, Alberta, Canada*.

Hu, Y., Palmé, T., & Fink, O. (2017). Fault detection based on signal reconstruction with auto-associative extreme learning machines. *Engineering Applications of Artificial Intelligence*, *57*, 105–117.

Jia, X., Karpatne, A., Willard, J., Steinbach, M., Read, J., Hanson, P. C., … Kumar, V. (2018, oct). *Physics Guided Recurrent Neural Networks For Modeling Dynamical Systems: Application to Monitoring Water Temperature And Quality In Lakes* (Tech. Rep.). Retrieved from

http://arxiv.org/abs/1810.02880

Julier, S. J., & Uhlmann, J. K. (1997). New extension of the Kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition vi* (Vol. 3068, p. 182). doi: 10.1117/12.280797

Khan, S., & Yairi, T. (2018, jul). *A review on the application of deep learning in system health management* (Vol. 107). Academic Press. doi: 10.1016/j.ymssp.2017.11.024

Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. In *3rd international conference on learning representations, iclr 2015 - conference track proceedings.*

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational bayes. In *2nd international conference on learning representations, iclr 2014 - conference track proceedings.*

LeCun, Y. A., Bottou, L., Orr, G. B., & Müller, K. R. (2012). Efficient backprop. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7700 LECTU*, 9–48. doi: 10.1007/978-3-642-35289-8-3

Michau, G., Hu, Y., Palmé, T., & Fink, O. (2017). Feature Learning for Fault Detection in High-Dimensional Condition-Monitoring Signals. *Submitted for a possible publication in IEEE Transactions on Cybenetics.*

Michau, G., Palmé, T., & Fink, O. (2017). Deep feature learning network for fault detection and isolation. In *Phm 2017, st. petersburg, usa, 2-5 october 2017* (pp. 108–118).

Moya, M. M., & Hush, D. R. (1996). Network constraints and multi-objective optimization for one-class classification. *Neural Networks*, *9*(3), 463–474. doi: 10.1016/0893-6080(95)00120-4

Nascimento, R. G., & Viana, F. A. (2019). Fleet prognosis with physics-informed recurrent neural networks. In *Structural health monitoring 2019: Enabling intelligent life-cycle health management for industry internet of things (iiot) - proceedings of the 12th international workshop on structural health monitoring* (Vol. 2, pp. 1740–1747).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Rausch, R. T., Goebel, K. F., Eklund, N. H., & Brunell, B. J. (2005). Integrated In-Flight Fault Detection and Accommodation: A Model-Based Study. In *Volume 1: Turbo expo 2005* (pp. 561–569). ASME. doi: 10.1115/GT2005-68300

Sarkar, D., Bali, R., & Ghosh, T. (2018). *Hands-on transfer learning with Python : implement advanced deep learning and neural network models using TensorFlow and Keras.*

Scheirer, W. J., de Rezende Rocha, A., Sapkota, A., & Boult, T. E. (2013). Toward Open Set Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(7), 1757–1772. doi: 10.1109/TPAMI.2012.256

Schölkopf, B., Williamson, R., Smola, A., Shawe-Taylor, J., & Piatt, J. (2000). Support vector method for novelty detection. In *Advances in neural information processing systems* (pp. 582–588).

Wang, Q., Michau, G., & Fink, O. (2019). Domain adaptive transfer learning for fault diagnosis. *arXiv preprint arXiv:1905.06004*.

Wang, S., Minku, L. L., & Yao, X. (2013). Online class imbalance learning and its applications in fault detection. *International Journal of Computational Intelligence and Applications*, *12*(04), 1340001.

Xu, L., Chow, M.-Y., & Taylor, L. S. (2007). Power distribution fault cause identification with imbalanced data using the data mining-based fuzzy classification e-algorithm. *IEEE Transactions on Power Systems*, *22*(1), 164–171.

Yucesan, Y. A., & Viana, F. A. C. (2019). Wind Turbine Main Bearing Fatigue Life Estimation with Physics-informed Neural Networks. In *Phm 2019* (Vol. 11, pp. 1–14). doi: 10.36001/PHMCONF.2019.V11I1.807

Zhang, Y., Li, X., Gao, L., Wang, L., & Wen, L. (2018). Imbalanced data fault diagnosis of rotating machinery using synthetic oversampling and feature learning. *Journal of manufacturing systems*, *48*, 34–50.

Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., & Gao, R. X. (2016, dec). *Deep Learning and Its Applications to Machine Health Monitoring: A Survey* (Tech. Rep.). Retrieved from http://arxiv.org/abs/1612.07640

Zhao, S., Song, J., & Ermon, S. (2019, jul). InfoVAE: Balancing Learning and Inference in Variational Autoencoders. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 33, pp. 5885–5892). doi: 10.1609/aaai.v33i01.33015885

Zhu, W., Miao, J., Qing, L., & Huang, G.-B. (2015). Hierarchical Extreme Learning Machine for unsupervised representation learning. In *2015 international joint conference on neural networks (ijcnn)* (pp. 1–8). IEEE. doi: 10.1109/IJCNN.2015.7280669

NOMENCLATURE

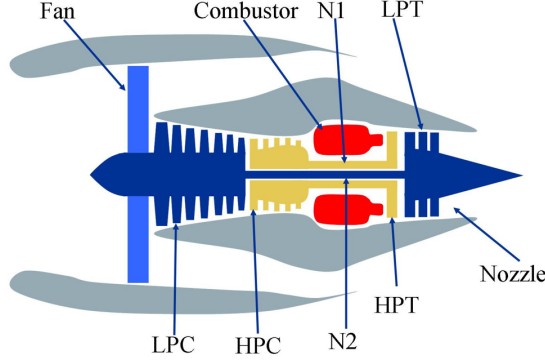|	
HPC	high-pressure compressor
HPT	high-pressure turbine
LPC	Low-pressure compressor
LPT	Low-pressure turbine
N1	Physical core speed
N2	Physical fan speed

Figure 18. Simplified diagram of the turbo fan engine model in C-MAPSS with Fan, Combustor, High-pressure compressor (HPC); high-pressure turbine (HPT); Low-pressure compressor (LPC); Low-pressure turbine (LPT); Physical core speed (N1), Physical fan speed (N2) and the Nozzle

9. APPENDIX I: NEURAL NETWORK OVERVIEW

In this section, we briefly introduce the selected discriminative and generative neural networks considered in our experiments. We focus first on discriminative models that try to learn $p(h_s|x)$ directly. In other words, algorithms that try to learn direct mappings from the space of inputs X or z to a label class (i.e. $\mathbf{T}$). In this group we introduce deep feed-forward networks (FF), vanilla autonecoders (AE) and hierarchical extreme learning machines (HELM). Finally, we will focus on generative algorithms that instead try to model the underlying distribution of the data $P(X)$ and show how these models can be combined with discriminative models to perform diagnostics tasks. In particular we introduce variational autoencoders (VAE).

9.1. Discriminative models

Feed-forward neural network (FF). A deep feed-forward (FF) neural network with L layers is a directed acyclic graph that implements a map $F : R^n \longmapsto R^{m^L}$ with the following structure:

$$F = F^L \circ F^{L-1} \circ \ldots \circ F^1 \tag{13}$$

$$F^l = \sigma^l \circ \overline{F}^l \tag{14}$$

$$\overline{F}^l(x) = W^l x + b^l \in R^{m^l} \tag{15}$$

Hence, a feed-forward neural network represents a family of functions $F_{\mathcal{H}}$ parameterized by parameters $\mathcal{H} = \{W^l, b^l\}_{l=1}^L$ (i.e. weight matrices W^l and biases b^l for each layer). σ_l denotes non-linear activation functions (e.g. tanh, ReLu, etc) and $\overline{F}^l$ denotes linear pre-activations. The number of neurons in each layer is given by m^l. We find the most appropriate function ($F_{\mathcal{H}}$) with the backpropagation algorithm (LeCun, Bottou, Orr, & Müller, 2012) given a training set $S_T = \{x^{(i)}, y^{(i)}\}_{i=1}^N$ of N input-output pairs.

The empirical risk on the training set S_T is generally selected as optimisation metric for generation of discriminative models. The empirical risk minimizer is defined as:

$$F_{\hat{\mathcal{H}}}(\mathcal{S}_T) = \arg \min_{\mathcal{H}} J(F_{\mathcal{H}}(x); \mathcal{S}_T) \tag{16}$$

where $\hat{\mathcal{H}}$ corresponds to the optimal weights and bias of the neural network F and $J(F; S_T)$ denotes the training risk of F on the training sample S_T

$$J(F; S_T) = \frac{1}{N} \sum_{i=1}^N \ell(y^{(i)}, F(x^{(i)})) \tag{17}$$

$$\ell(y, F(x)) = \frac{1}{2}||y - F(x)||_2^2 \tag{18}$$

and the output target label y corresponds to: $y = \mathbf{T}$ for the one-class network.

Autonencoders (AE). An autoencoder is any neural network that aims to learn the identity map (i.e. it is trained to reconstruct its own input). Therefore, it is a special case of the previous networks consisting of two parts with symmetric topology: an encoder (E) and a decoder (D). The encoder provides an alternative representation of the input (x) that we denote as z and the decoder reconstructs back the input (i.e. $\overline{x}$) as closely as possible from its encoded representation z. The resulting mapping corresponds to the following structure:

$$F = D \circ E \tag{19}$$

$$E = F^{l_z} \circ \ldots \circ F^1 : \mathbb{R}^n \to \mathbb{R}^d, x \to z : x^l \tag{20}$$

$$D = F^L \circ \ldots \circ F^{l_z+1} : \mathbb{R}^d \to \mathbb{R}^n, z \to \overline{x} \tag{21}$$

where the layer l_z is generally a bottleneck (i.e. $d < n$) and therefore z is a compressed representation of the input. Autoencoders can lean powerful non-linear generalization of principal component analysis (PCA).

The loss function of autoencoders is

$$\ell(x, \overline{x}) = \frac{1}{2}||x - F(x)||_2^2 \tag{22}$$

Hierarchical Extreme Learning Machines (HELM). Hierarchical Extreme Learning Machines are another popular neural network class for diagnostics task. Several researches have shown that it outperforms traditional machine learning method such us PCA and SVM in diagnostics task (Michau, Hu, et al., 2017). HELM networks share similarities to three methods described earlier but with different topology and training method. As in deep RNN and FF networks, a HELM of L layers has a hierarchy representations levels at each layer (i.e. s^l. This hierarchical hidden state s^l that evolves as a function of the previous state s^{l-1} defining a directed acyclic graph.

However, in this case it evolves as a linear transformation.

$$s^l = G^l(s^{l-1}, \beta^l) \in R^{m^l} \tag{23}$$

$$G^l(s^{l-1}, \beta^l) = s^{l-1}\beta^{l^T} \quad l = 1, \ldots, L-1 \tag{24}$$

with $s^0 := x$

The output of a HELM network is connected to the state of the last hidden layer s^{L-1} as follows:

$$F = F^L \beta^L \tag{25}$$

$$F^L = \sigma^L \circ \overline{F}^L \tag{26}$$

$$\overline{F}^L = W^L s^{L-1} + b^L \tag{27}$$

$$s^{L-1} = G^{L-1} \circ \ldots \circ G^1(x) \tag{28}$$

Contrarily to previous networks the parameters $\mathcal{H} = \{W^l, b^l\}_{l=1}^L$ (i.e. weight matrices W^l and biases b^l for each layer) are random and are not optimised. Therefore, they provide an alternative (random) representation of the state s^{l-1} (i.e. F^l) given weights $\{W^l, b^l\}$ and the non-linear transformation σ^l. The weight matrix β^l are optimised layer wise to reconstruct the state s^{l-1} from this random projection. Therefore, the loss function of β resembles the auto encoder loss. However, typical regularization schemes are required correspond to the Maximum at Posterior (MAP)

$$\beta^l = \arg\min_{\beta^l} \lambda||\beta^l||_1 + ||F^l\beta^l - s^{l-1}||_2^2 \tag{29}$$

$$F^l = \sigma^l \circ \overline{F}^l \tag{30}$$

$$\overline{F}^l = W^l s^{l-1} + b^l \tag{31}$$

with $s^L := y$

HELM are typically referred as autoencoder network due to the training process of the network, where the weight matrix β is obtain from solving an autoencoder network for each of the hidden layers of HElM.

9.2. Generative models

Contrarily to the discriminate models that try to learn $p(h_s|x)$ directly, generative algorithms model the underlying distribution of the data $p(x)$. Concretely, generative latent models assume that an observed variable x is generated by some random process involving an unobserved random (i.e. latent) variable z (Sarkar, Bali, & Ghosh, 2018). Hence, latent models define a joint distribution $p(x, z) = p(x|z)p(z)$ between a feature space z, and the input space x (S. Zhao, Song, & Ermon, 2019). Hence, the underling generation process resort to two steps: 1) a value $z^{(i)}$ is generated from some prior distribution $p(z)$ and 2) a value $x^{(i)}$ is generated from some conditional distribution $p(x|z)$. Hence, the data generation process is modeled with a complex conditional distribution $p_\theta(x|z)$, which is often parameterized with a neural network.

There are two big families of generative models: generative adversarial networks (GANs) and Variational Autoencodes (VAEs). Our proposed method is based on VAEs that we explained in the following.

Variational Autoencoder (VAE). Variational autoencoders (Kingma & Welling, 2014) aim to sample values of z that are likely to have produced x and compute $p(x)$ from those (Doersch, 2016). As in the case of the standard vanilla autonocodes, VAE models comprise of an inference network (or encoder) and a generative network (or decoder). Contrarily to previews models, the latent representation z of the data x is a stochastic variable. Therefore, the encoder and the decoder networks are probabilistic. The inference network $q_\phi(z|x)$, parametrizes the intractable posterior $p(z|x)$ and the generative network $p_\theta(x|z)$ parametrizes the likelihood $p(x|z)$ with parameters θ and ϕ respectively. These parameters are the weights and biases of the neural network. A simple prior distribution $p(z)$ over the features is generally assumed (such us Gaussian or uniform).

The natural training objective of a generative model is to maximize the marginal likelihood of the data

$$\mathbb{E}_{p(x)}[\log p_\theta(x)] = \mathbb{E}_{p(x)}[\mathbb{E}_{p(z)}[\log p_\theta(x|z)] \tag{32}$$

However, direct optimization of the likelihood is intractable since $p_\theta(x) = \int_z p_\theta(x|z)p(z)dz$ requires integration (S. Zhao et al., 2019). Therefore, VAE consider the an approximation to the marginal likelihood denoted Evidence Lower BOund or ELBO; which is a lower bound to the log likelihood

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{p(x)}[\mathbb{E}_{q_\phi(x|z)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p(z))] \tag{33}$$

$$\leq \mathbb{E}_{p(x)}[\log p_\theta(x)] \tag{34}$$

where D_{KL} denotes the Kullback-Leibler. Hence, the training objective of VAE is to optimize the lower bound with respect to the variational parameters ϕ and the generative parameters θ

$$\max_{\phi, \theta} \mathbb{E}_{p(x)}[\mathbb{E}_{q_\phi(x|z)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p(z))] \tag{35}$$

The ELBO objective can be viewed as the sum of two components. The first term is the expected negative reconstruction error and it is similar to the training objective of a vanilla autoencoder. The KL divergence ($D_KL \geq 0$) is a distance measure of two probability distributions and acts as a regularizer of ϕ trying to keep the approximate posterior $q_\phi(z|x)$ close to the prior $p(z)$.

Under certain hypothesis on the distribution families the KL divergence can be integrated analytically and therefore only the expected reconstruction error requires estimation by sampling. Therefore, direct optimization of $\mathcal{L}_{\text{ELBO}}$ with the back-

propagation algorithm requires a good estimate of the gradient of the expectation $\nabla_\phi \mathbb{E}_{q_\phi(x|z)}[\log p_\phi(x|z)]$. However, naive Monte Carlo estimators exhibit very large variances and are therefore impractical. To find a low-variance gradient estimator a reparametrization of z with a differentiable transformation $z = g(\epsilon, x)$ of an auxiliary noise variable ϵ is introduced (Kingma & Welling, 2014). The function $g(x, \epsilon)$ is generally chosen that maps an input datapoint $x^{(i)}$ and noise vector ϵ to a sample from the approximate posterior. The sampled $z^{(i)}$ is then input to the function $\log p_\theta(x|z)$ providing probability mass of a data point under the generative model p_θ. Figure 19 shows the resulting network architecture. As a
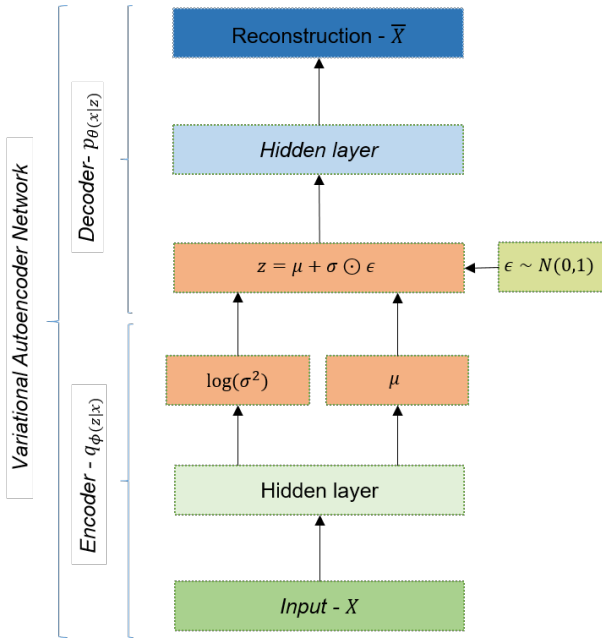

Figure 19. Variational autoencoder network

be computed analytically

$$- D_{KL}(q_\phi(z|x)||p(z)) =$$

$$\sum_{j=1}^{d}(1 - \log(\sigma_j^{(i)})^2 - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2) \quad (37)$$

default assumption in VAE, the variational approximate posterior $q_\phi(z|x)$ follows a mutivariate Gaussian with diagonal covariance (i.e $q_\phi(z|x) = \mathcal{N}(z; \mu, \sigma^2 \mathbf{I})$). This assumption arises from the hypothesis that the true but intractable posterior $p_\theta(z|x)$ takes also the shape of an approximate Gaussian form with diagonal covariance. The distributions parameters of the approximate posterior μ and $\log \sigma^2$ are the non-linear embedding of the input x provided by the encoder network with variational parameters ϕ. Hence, the encoder output is a paramentrization of a approximate posterior distributions. Under these assumptions a valid local reparametrization of z that allows to sample from the assumed Gaussian approximate posterior (i.e. $z^{(i)} \sim q_\phi(z|x^{(i)})$) is

$$z^{(i)} = \mu^{(i)} + \sigma^{(i)} \odot \epsilon \quad (36)$$

with $\epsilon \sim \mathcal{N}(0, \mathbf{I})$.

Since in this model we assume that both $p_\theta(z)$ and $q_\phi(z|x)$ are Gaussian distribution and therefore the $D_{KL}(q_\phi(z|x)$ can