A New Multivariate Approach for Prognostics based on Extreme Learning Machine & Fuzzy Clustering

Kamran Javed, Rafael Gouriveau, *Member, IEEE,* Noureddine Zerhouni, *Member, IEEE*

Abstract—**Prognostics is a core process of Prognostics & Health Management (PHM) discipline, that estimates the Remaining Useful Life (RUL) of a degrading machinery to optimize its service delivery potential. However, machinery operates in a dynamic environment and the acquired condition monitoring data are usually noisy and subject to high level of uncertainty / unpredictability, which complicates prognostics. The complexity further increases, when there is absence of prior knowledge about ground truth (or failure definition). For such issues, data-driven prognostics can be a valuable solution without deep understanding of system physics. This paper contributes a new data-driven prognostics approach namely, an "enhanced multivariate degradation modeling", which enables modeling degrading states of machinery without assuming a homogeneous pattern. In brief, a predictability scheme is introduced to reduce the dimensionality of the data. Following that, the proposed prognostics model is achieved by integrating two new algorithms namely, the Summation Wavelet-Extreme Learning Machine & Subtractive- Maximum Entropy Fuzzy Clustering to show evolution of machine degradation by simultaneous predictions and discrete state estimation. The prognostics model is equipped with a dynamic failure threshold assignment procedure to estimate RUL in a realistic manner. To validate the proposition, a case study is performed on turbofan engines data from PHM challenge 2008 (NASA), and results are compared with recent publications.**

Index Terms—**Data-driven, Extreme Learning Machine, Fuzzy clustering, Prognostics, RUL.**

ABBREVIATIONS

b	Bias.
c	Clusters.
C_L	Unsupervised classifier.
C_m	Matched classifier.
f	Activation function.
F_L	Multidimensional features data.
H	Prediction horizon.
H_{avg}	Hidden layer output matrix.
I	RUL error interval.
L	Performance limit.
L_D	Learning data.
M	Prediction Modeling tool.
msp	Multi-step ahead prediction.
N	Learning data samples.
$Pred$	Predictability.
P	Univariate predictor.
tc	Current time.
tf	Failure time.
TF	Test data set.
v	Cluster centers.
w	Weights.
x, t	Model input and target.

I. INTRODUCTION

With aging, machinery or its components are more vulnerable to failures. The optimization of machinery service and the minimization of life cycle costs / risks require continuous monitoring of deterioration process, and reliable prediction of life time at which it will be unable to perform desired functionality. To fulfill such time critical needs, PHM is an emerging engineering discipline which links studies of failure mechanisms and life cycle management [1]. It aims at extending machinery service life, while reducing exploitation and maintenance costs. PHM has by and large been accepted by the engineering systems community in general and the aerospace industry in particular as the future direction [2].

Within the framework of PHM, prognostics is considered as a key process with future capabilities, which should be performed efficiently for successful decision support to recommend actions for maintenance [3], or system configuration [4]. Prognostics is called as the "estimation of a system's Remaining Useful Life", as its primary objective is to intelligently use monitoring information of in-service machinery, and to predict its RUL before a failure occurs, given the current machine condition and its past operation profile [5], [6]. RUL is expressed by considering units corresponding to the primary measurement of use for overall system, e.g. for commercial aircrafts the unit is cycles, for automobiles it is kilometers.

According to literature, prognostics approaches are classified into three types: physics based, data-driven and their hybrid [7]. The physics based approaches for prognostics require explicit mathematical model to describe the behavior of a degrading system and its components [8] and are thereby application specific [9]. Data-driven prognostics approaches learn the systems behavior directly from collected condition monitoring (CM) data and use that knowledge to infer its current state and to predict future progression of failure to estimate RUL. Hybrid approaches integrate physics based and data-driven approaches, that attempts to leverage the strengths from both prognostics approaches. But in parallel, such methods are computationally expensive, which makes them difficult for some applications.

Although in recent years, a vast number of prognostics methods have been proposed for different applications, the progress to build an effective and efficient approach is still limited. Usually, lack of understating about complex and non-linear behavior of degrading machinery under dynamic operating

environment prevents practitioners to develop precise mathematical models for prognostics. In such situations, data-driven prognostics is a good alternative and is easier to deploy without physical understanding about the degradation process. We can classify RUL estimation strategies of data-driven approaches into three groups (Fig. 1).
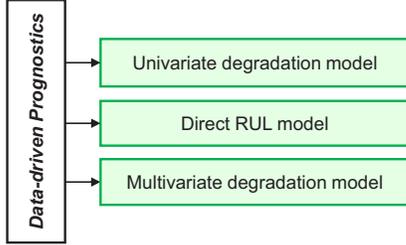
Fig. 1. Data-driven RUL estimation strategies

- Univariate degradation based prognostics model rely on the prediction of continuous degrading state followed by a failure criteria. RUL estimate is obtained when degrading signal intersects a pre-defined failure threshold (FT). The main difficulty with this method is to define FTs, which is often very hard to achieve.
- Direct RUL prediction model learns from the data, the relation between observed trends and equipment end of life. RUL is derived from data-driven model by applying a pattern matching process between the current observation and the knowledge of equipment RUL [10]. This method does not require FTs, but rely on smooth and monotonic features for pattern matching [11].
- Multivariate degradation based prognostics model is composed of two complementary modules: a prediction engine that forecasts observations in time, a classifier that sets precise FT and estimates the most probable states of degradation [12]. RUL is obtained from the estimated time to reach the faulty state from current time tc. This idea was initially proposed in [13]. A complete illustration of this method was given in [14], [15]. Later, this approach was further developed and named as joint approach for RUL [16].

Obviously no model is perfect, however, among data-driven prognostics strategies, the multivariate degradation approach is relatively new as compared to former methods (see Fig. 2). Also, to improve the accuracy of RUL estimates, the use
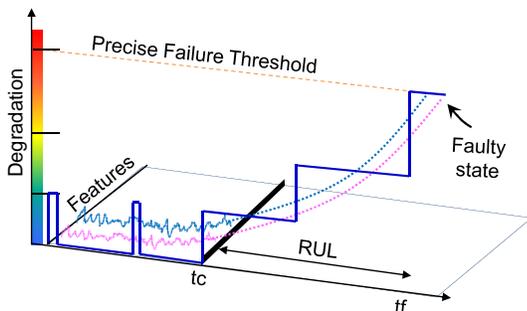
Fig. 2. Simultaneous prediction & discrete state estimation

of multidimensional degradation signals is preferred rather than one-dimension signal [16]. But still, some issues of this approach should be addressed to improve prognostics. Firstly, the prediction phase do not account for the selection of most relevant features that are predictable over long horizon and can contribute for better RUL estimation. Secondly, the classification phase is based on the assumption about the number of degrading states. In case of the absence of ground truth (or actual labels) in the data, clustering method [17] or a posterior classification [14] are applied, and the number of states is same for all machines in the fleet. As a result, the FTs are static and do not vary with respect to test case, which is not realistic. Therefore, the need for an enhanced prognostics approach can be pointed out to avoid issues highlighted above. This paper contributes to an "enhanced multivariate degradation prognostics", that is as achieved by the following developments.

1) Data post-treatment: in order to reduce the dimensionality of data, a predictability based features selection procedure is proposed. The underlying idea is to assess the capability of a prediction model to predict a feature. Obviously, there is no interest in retaining features that are hard to be predicted.
2) Prediction model: in order to achieve long-term predictions, a new connectionist approach is presented namely, the Summation Wavelet-Extreme Learning Machine (SW-ELM) algorithm, that enables a good balance between model accuracy and complexity.
3) Classification model: in order to assign temporal predictions to different classes, a new clustering algorithm is presented namely, the Subtractive-Maximum Entropy Fuzzy Clustering (S-MEFC), that enables estimating the health state of the system automatically, (rather than with fixed no. of states).
4) Dynamic FT: to estimate the RUL, SW-ELM and S-MEFC are integrated in order to dynamically set the FTs and to track the evolution of machine degradation with simultaneous predictions and discrete state estimation.

Due to the contributions above, the proposed prognostics approach assesses the health state of machinery dynamically and estimates its RUL in a realistic manner. Moreover, the combination of methods like SW-ELM and S-MEFC is a less explored area, but it has a good potential to overcome drawbacks of conventional data-driven approaches. Like other data-driven methods the combination of such machine learning methods can be deployed quickly and cheaply.

The paper is organized as follows. Section II focuses on improving multivariate degradation modeling by proposing a predictability measure and algorithms for prediction / classification. Section III elaborates the dynamic FT procedure to complete our proposition. Section IV demonstrates performance of proposed approach on a PHM case study on turbofan engines data (NASA). Finally, section V concludes this work.

II. Enhanced multivariate prognostics modeling

A. Improving the prediction phase

Describing the problem of prognostics by means of continuous predictions and discrete state estimations is new in PHM community and its only addressed in few publications [14], [16], [18]. Obviously, for prognostics part, the prediction phase is critical and must be dealt in an accurate manner for timely decisions. However, due to dynamic operational environment, the CM data gathered from machinery are subject to high levels of uncertainty and unpredictability. Predictions can be incorrect; therefore, it is important to understand their quality in a framework that is dependent on the considered time series. In other words, for further classification task (to estimate discrete states), there is no use in retaining features that are not predictable. Moreover, the accuracy of prediction is greatly affected by horizon of prediction and the choice of prediction model. According to above discussions, we can highlight two key issues of the prediction phase.

- How to asses predictability of features?
- How to achieve long-term predictions?

To improve the prediction performances, it is required to include the data post-treatment procedure for reduction of multidimensional data and to build an efficient tool to project the evolution of degradation process.

1) Predictability based features selection: the standard of machine learning approaches for prognostics is composed of two steps, i.e., learning and testing. In classical way, the learning step has to be reiterated by trial and error, since there is no certainty in providing an accurate prognostics model. This can be time consuming, as there is no advantage in retaining features that are hard to be predicted. Therefore, while building a prognostics model the learning step should include set of features that can be more accurately predicted over different horizons. In this context the considered post-treatment approach emphasizes to further reduce dimensionality of data by assessing predictability. According to [18], predictability is defined as follows.

- Predictability is "the ability of a given time series TS to be predicted with an appropriate modeling tool M that facilitates future outcomes over a specific horizon H and with a desired performance limit L".

Formally, we propose to formulate it as:

$$Pred(TS/M, H, L) = exp^{-\left|ln\left(\frac{1}{2}\right).\frac{MFE_{TS/M}^{H}}{L}\right|} \quad (1)$$

where, MFE is the mean forecast error between the actual values of TS and the predicted ones (see details in [18]). Predictability has an exponential form (Fig. 3) and is inversely proportional to MFE. A TS can be considered predictable if predictability coefficient is between 0.5 and 1. The aim of this measure is to reconsider the learning step of data-driven approach by considering both "feature selection" and "prognostics modeling" as interrelated. This allows to either retain / reject each potential couple of "feature-model" to be used for prognostics. On this basis, the set of selected features obtained by data-mining techniques can be further reduced to a final set of predictable features in accordance to trained prediction models.
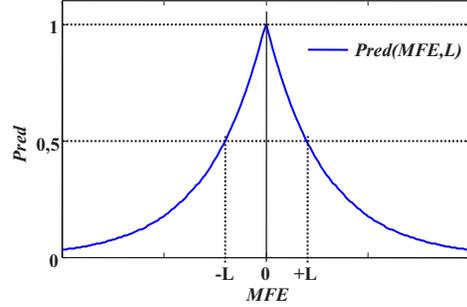
Fig. 3. Illustration of predictability measure

The predictability based feature selection procedure assumes that different prediction models are available e.g. artificial neural networks (ANN), Neuro-fuzzy systems, etc. However, a feature can be unpredictable by a classical ANN, but may be predictable by NFS. Indeed, its hard to build an accurate prediction model and the possibility exists that considered prediction models are incapable of predicting the most useful features. Therefore, in the learning phase the practitioner should carefully choose the prediction approach for a particular application, which is the aim of the following topic (see [18] for further details).

2) Building a prediction model using SW-ELM: in PHM applications, prognostics approaches are used to compute long term predictions followed by FT. The accuracy of a prognostics model is related to its ability to predict degrading behavior of a monitored machinery. Among machine learning methods, Artificial neural networks are a special case of adaptive networks that are most commonly used in PHM domain [7], [19]. They do not require any *a priori* assumptions on the properties of given data set, neither they assume any *a priori* model [20]. ANNs can have two types of architectures known as feed-forward neural networks (FFNs) and recurrent neural networks. It is mentioned that around 95% of literature are on FFNs [21]. However, constructing a good neural network model is a non-trivial task and practitioners still have to face several issues that limit their applicability for real industrial case [22]. Although, several learning schemes for ANNs exist [23], relatively a new algorithm was proposed for a Single Layer Feed Forward Networks (SLFN), namely the Extreme learning Machine (ELM) [24].

Unlike conventional learning algorithms for SLFN, ELM avoids slow iterative tuning and requires one-pass batch learning, by randomly choosing hidden layer parameters (weights and bias) and analytically calculating output layer weights. ELM has several other advantages over classical methods of ANNs, which makes it a better candidate for prognostics [25]. Moreover, ELM has been proved for its universal approximation capability [26]. But, obviously, no algorithm is perfect, the performances of ELM can suffer due to random initialization of parameters and the type of activation functions in hidden layer. Such issues can increase the complexity of model and may lead to ill-condition [25]. In other words,

accuracy of ELM is sensitive to the complexity of hidden layer and usually a large model is generated [27]. Therefore, to enhance the performances of ELM without compromising its better applicability, an improved variant is presented namely, the Summation Wavelet-Extreme Learning Machine.

Basically, SW-ELM is a combination of ANN and wavelet theory and appears to be an effective prediction approach [28]. It benefits from an improved: parameter initialization to minimize the impact of random weights and bias (of input-hidden layer) and structure with dual activation functions for a hidden node. Also SW-ELM works on actual scales of the data (see Fig. 4). The differences with ELM are further elaborated as follows.

- Structure - Each hidden node holds a parallel conjunction of two different activation functions rather than a single activation. Output from a hidden neuron is the average value from dual activations ($\bar{f} = (f_1 + f_2)/2$). The conjunction of activation functions can handle non-linearity in a better manner (rather than single activation function), which has been confirmed in our recent publication [28] and in the previous studies by [29].
- Activation function - Convergence of algorithm is improved by an inverse hyperbolic sine (Eq. 2) and a Morlet wavelet (Eq. 3) as dual activation functions.

$$f_1 = \theta(X) = log\left[x + (x^2 + 1)^{1/2}\right] \quad (2)$$

$$f_2 = \psi(X) = cos(5x)e^{\left(-0.5x^2\right)} \quad (3)$$

- Parameter initialization - To provide a better starting point to the algorithm, two types of parameters have to be considered: those from the wavelets (dilation and translation) adapted by a heuristic procedure such that the wavelet function covers the whole input space [30], and those from the $SLFN$ (weights & bias for input-hidden layer), initialized by Nguyen Widrow procedure [31].
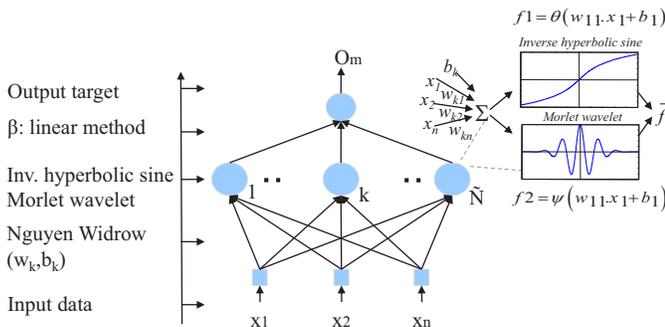
Fig. 4. Machine learning view of SW-ELM

Let note n and m the numbers of inputs and outputs, N the number of learning data samples (x_i, t_i), where $i \in [1 \ldots N]$, $x_i = [x_{i1}, x_{i2}, ..., x_{in}]^T \in \Re^n$ and targets $t_i = [t_{i1}, t_{i2}, ..., t_{im}]^T \in \Re^m$, and $\tilde{N}$ the number of hidden nodes, each one with activation functions (f_1 and f_2). For each sample j, the output o_j is mathematically expressed as:

$$\sum_{k=1}^{\tilde{N}} \beta_k \bar{f}\left[(\theta, \psi)(w_k.x_j + b_k)\right] = o_j \ , j = 1, 2, ..., N \quad (4)$$

where $\bar{f}$ is the average output from two different activation functions θ and ψ. $w_k = [w_{k1}, w_{k2}, ..., w_{kn}]^T \in \Re^n$ is an input weight vector connecting the kth hidden to input layer neurons, $(w_k.x_j)$ is the inner product of weights and inputs, and $b_k \in \Re$ is the bias of kth neuron of hidden layer. Also, $\beta_k = [\beta_{k1}, \beta_{k2}, ..., \beta_{km}]^T \in \Re^m$ is the weight vector to connect the kth neuron of hidden layer and output neurons. To minimize the difference between output o_j and target t_j, $\sum_{j=1}^{\tilde{N}} \|o_j - t_j\| = 0$, there exist β_k, w_k and b_k such that:

$$\sum_{k=1}^{\tilde{N}} \beta_k \bar{f}\left[(\theta, \psi)(w_k.x_j + b_k)\right] = t_j \ , j = 1, 2, ..., N \quad (5)$$

which can be expressed in matrix form as,

$$H_{avg}\beta = T \quad (6)$$

where H_{avg} is a $\left[N \times \tilde{N}\right]$ matrix expressed as,

$$H_{avg}(w_1, \ldots, w_{\tilde{N}}, x_1, \ldots, x_{\tilde{N}}, b_1, \ldots, b_{\tilde{N}}) =$$

$$\bar{f}(\theta, \psi)\begin{bmatrix} (w_1.x_1 + b_1) & \ldots & (w_{\tilde{N}}.x_1 + b_{\tilde{N}}) \\ \vdots & \ldots & \vdots \\ (w_1.x_N + b_1) & \ldots & (w_{\tilde{N}}.x_N + b_{\tilde{N}}) \end{bmatrix} \quad (7)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix}_{\tilde{N} \times m} \quad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m} \quad (8)$$

Finally, the least square solution of the linear system defined in Eq. 6, with minimum norm of output weights β is:

$$\hat{\beta} = H_{avg}^{\dagger}T \quad (9)$$

where $H_{avg}^{\dagger}$ shows the Moore-Penrose generalized inverse for the hidden layer output matrix H_{avg} [32]. A brief learning scheme is synthesized in Algo. 1, (see details in [28]).

Algorithm 1 Brief learning scheme of SW-ELM

Require
- N learning samples (x_i, t_i), n inputs, $\tilde{N}$ hidden nodes
- Arcsinh and Morlet activation functions (θ and ψ)

Ensure
- Initialize weights, bias and Morlet wavelet parameters
- Find output weights matrix β

1: Initialization of wavelet parameters (dilatation and translation), by [30].
2: Assign parameters of hidden nodes i.e., weights and bias (w_k, b_k), by Nguyen Widrow approach [31].
3: Obtain hidden layer output matrix H_{avg} using Eq. 7 .
4: Find the output weight matrix $\hat{\beta}$ in Eq. 9.

---

3) Long-term predictions: Connectionist approaches like SW-ELM aim at approximating an input-output function. Such methods must be tuned to fit to the studied problem. The long-term multi-steps ahead prediction "msp" can be achieved in different ways and by using connectionist approaches (structure + learning algorithm). [33] studied five methods, namely the Iterative, Direct, DirRec, Parallel, and MISMO approaches.

According to this work, except iterative approach all other approaches require knowledge of prediction horizon H. The msp are provided by using a single model (SW-ELM) that is tuned to perform a one-step ahead prediction $\hat{x}_{t+1}$. This estimated value is used as the regressors of the model to estimate the following ones and the process is repeated until the estimation of $\hat{x}_{t+H}$. The procedure is illustrated in Fig. 5. Formally:

$$\hat{x}_{t+h} = \begin{cases} \text{if } h = 1, f^1\left(x_t, \ldots, x_{t+1-p}, [\theta^1]\right) \\ \text{elseif } h \in \{2, \ldots, p\}, \\ f^1\left(\hat{x}_{t+h-1}, \ldots, \hat{x}_{t+1}, x_t, \ldots, x_{t+h-p}, [\theta^1]\right) \\ \text{elseif } h \in \{p+1, \ldots, H\}, \\ f^1\left(\hat{x}_{t+h-1}, \ldots, \hat{x}_{t+h-p}, [\theta^1]\right) \end{cases} \quad (10)$$

where $\left\{P^1, [\phi^1]\right\}$ states for the one-step ahead prediction model and its parameters obtained from the learning step, p the number of regressors used. Note that, from the time $h > p$, predictions are made only on evaluated data and not on observed data.
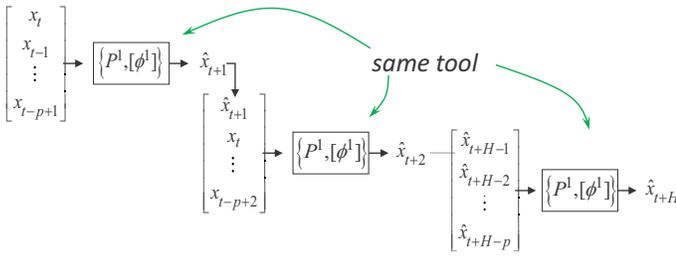
Fig. 5. Iterative model for multi-steps predictions [33]

As discussed above, the iterative approach do not need knowledge of H and require only one model to perform msp. However, during the iterative process, the prediction model takes the estimated values as inputs rather that the actual ones, which will affect subsequent predictions.

In order to address this issue, given the msp problem, in the learning phase several SW-ELM models can be obtained for the considered data. The model with minimum training error is chosen for performing msp in the test phase. Nevertheless, the learned SW-ELM model may not be optimum, therefore to ensure that error of msp is acceptable for the given feature data, predictability analysis is performed (see II-A1).

B. Improving the classification phase

With multivariate degradation modeling, it is not only required to perform good predictions, but to stop predictions at the right time as well. Therefore, the task to determine the prediction horizon is performed by a classifier that assesses the health state of the monitored machinery using multi-dimensional data and sets the FTs. For PHM applications, failure definitions could be a difficult task to achieve. The uncertainty of RUL estimates is not only due to predictions, but to FTs as well, that can lead to large RUL errors. Also, prognostics can be quite challenging when few knowledge or previous experiences on degradation process are available [34], or when there is absence of ground truth. In other words, multivariate degradation based prognostics can be difficult (section I), as one has to handle unlabeled data with no prior knowledge about actual states or transitions between these states. According to above discussions, we can highlight two key issues of the classification phase.

- How to determine states of degrading machinery?
- How to handle unlabeled data?

To enhance multivariate degradation modeling, an unsupervised classification procedure is necessary. Moreover, it is also important to represent uncertainty of unlabeled multidimensional data like, transitions among different states, which can be performed by theories like fuzzy sets, probability, etc. Therefore, we consider a new algorithm, that accounts for issues above and can improve prognostics model efficiency.

1) Building a classifier using S-MEFC: the goal of unsupervised classification is to organize data into homogeneous groups such that, intra group similarity is minimized to have compact clusters and inter group dissimilarity is maximized to increase separation among clusters. As the CM data for prognostics are dynamic and vary with time. Given unlabeled multidimensional real valued data, it is required to determine groups of time series of unequal lengths, to assess the health of monitored machinery. According to literature, clustering methods can be classified into five categories: model based methods, graph based methods, density based methods, hierarchical methods and partition methods [35]. The clustering methods are totally data-driven and each approach can have its own limitations, however some key issues can be: sensitivity to noise, inconsistent performance (i.e., results vary for each run), time consumption and dependence on human involvement. Therefore, a recently proposed unsupervised classification algorithm is considered namely, the Subtractive-Maximum Entropy Fuzzy Clustering [12] that enables estimating the health state of the system. S-MEFC algorithm takes benefits of density based Subtractive Clustering algorithm (SC) [36] and partition based Maximum Entropy Fuzzy Clustering algorithm (MEFC) [37]. S-MEFC is summarized in Algo. 2. In brief:

- SC is a one pass approach to estimate cluster centers among numeric data (with a density function). This method can be used to provide number of clusters and initial centers for algorithms like MEFC [37]. According to [38], SC is a robust algorithm that could (a) remove outliers, (b) resolve conflicting data and (c) discard unnecessary patterns [36]. Also, it does not have inconsistency issue like Fuzzy C-Means algorithm that gives different results for different simulations [39]. However, it does not represent uncertainty of clustered data.
- MEFC algorithm is considered to represent uncertainty of clustering phase, as it ensures a maximum fairness to handle imprecise data and minimize such choice of bias of membership function via maximum entropy inference (MEI). In comparison to other fuzzy clustering approaches, the maximum entropy function also gives a clear physical meaning for clustering data. This means, the data points closer to cluster centers will have higher memberships.

Consider a training dataset given by Eq. 11 containing N unlabeled samples of $\ddot{n}$ time series. SC approach is used to automatically determine clusters c in multidimensional data,

and their centers $V = \{v_j\}_{j=1}^c$ (see [36]).

$$L_D = \begin{bmatrix} x_{11} & \ldots & x_{1\ddot{n}} \\ \vdots & \ldots & \vdots \\ x_{N1} & \ldots & x_{N\ddot{n}} \end{bmatrix} \quad (11)$$

SC requires one parameter which is set by the user, i.e., the radius of neighborhood ra. The achieved centers, V from SC serve the need of MEFC algorithm, which avoids its random initialization of centers. To further adjust centers positions and to assign membership to each data point, given σ a fuzziness parameter (by user), the algorithm runs iteratively until termination criteria (ϵ) is met. The maximum entropy inference based fuzzy partition matrix is represented as $U = [\mu_{ij}]_{c \times N}$, where μ_{ij} represents the membership degree of ith object in j^{th} cluster. Note that, the key component in clustering is similarity between two data being compared [35]. In our case we consider Standardized Euclidean Distance D_{SE}, while updating cluster partitions matrix U and centers matrix V with MEFC algorithm. It is similar to Euclidean Distance (ED) except that every dimension is divided by its standard deviation. This results in better clustering than would be achieved with ED, because each dimension has different scales. Let x, v each be a $\ddot{n}$-dimensional vector and SD the standard deviation. The D_{SE} distance between data points and the centers is computed as:

$$D_{SE}(x,v) = \sqrt{\sum_{k=1}^{\ddot{n}} \left(1/SD_k^2\right)(x_k - v_k)^2} \quad (12)$$

Algorithm 2 S-MEFC

Require   - Learning data set Eq. 11
            - Fix ra, ϵ, $\sigma > 0$
Ensure   - Cluster centers V and fuzzy partitioning U

learning procedure

1:    Obtain initial cluster centers v^{old} using SC ([36])
2:    Compute fuzzy partition matrix U using MEI ([37])

$$\mu_{ij} = \frac{e^{-D_{SE_{ij}}^2/2\sigma^2}}{\sum_{k=1}^c e^{-D_{SE_{ik}}^2/2\sigma^2}} \quad \forall i,j \quad (13)$$

3:    Adjust cluster centers v^{new}

$$v_j^{new} = \frac{\sum_{i=1}^N \mu_{ij}.x_i}{\sum_{i=1}^N \mu_{ij}} \quad \forall j \quad (14)$$

4:    Repeat step 2 and 3 until termination criteria is met

$$\left\| v^{new} - v^{old} \right\| < \epsilon \quad (15)$$

---

2) Discrete state estimation: for prognostics model, the discrete state estimation is achieved by two steps, 1) off-line and 2) on-line. The off-line step is required to determine the hidden structure (in the multidimensional data) representing states of degradation, i.e., good state to faulty state. This step can be referred as the learning phase of an unsupervised classifier. The on-line step deals with labeling new data in test phase. Therefore, labels (or discrete states) are assigned by looking at similarity between multidimensional time series and clusters achieved in the off-line phase. To illustrate this procedure, consider a 2D visualization is shown in Fig. 6. The multidimensional time series data are firstly assessed by the unsupervised classifier to determine probable states of degradation (i.e., transition from good state to faulty state), and in the test phase new data from predictions are matched with obtained clusters. On the basis of similarity metric, data closer to particular cluster centers are given labels that represent the discrete states (S), and the RUL estimate is achieved when a transition from degrading (d) state to faulty (f) occurs, indicating a FT to stop prediction process (see Eq. 16).
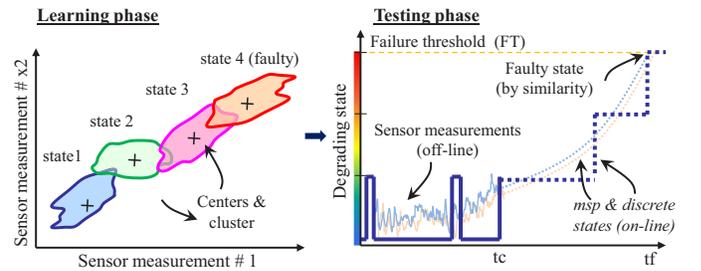
Fig. 6. Discrete state estimation from multidimensional data

$$transition\ d \overset{State}{\rightarrow} f \Rightarrow RUL = S_{d \rightarrow f} - tc \quad (16)$$

III. DYNAMIC FAILURE THRESHOLDS PROCEDURE

Generally, modeling of data-driven prognostics with machine learning methods requires learning and testing steps, that are dependent on features extracted / selected from raw CM data. As discussed in section I, for multivariate degradation modeling, prognostics is performed by integrating a prediction model and a classification model to perform simultaneous predictions and discrete state estimation (or health assessment) to achieve the RUL estimate. However, with this approach, previous works presented in [14], [16], [18] have a common drawback that the number of discrete states are assumed for continuous observations. This assumption is not realistic in case of degrading machinery. The behavior of each machine can differ according to operational environment. For practical reasons such assumptions are avoided. Therefore, during operation, each machine can have different states or degradation levels toward failure [40], [41]. As a result, FTs can vary and should be assigned dynamically, rather than static ones based on fixed number of states as presented in [16]. To achieve the goal of dynamic FTs, we propose a new procedure that accounts for issues related to FTs and improves the performance of the prognostics model.

The proposed procedure integrates two new algorithms namely, the SW-ELM and the S-MEFC that enables estimating the RUL of degrading machinery. The main task of SW-ELM based predictor is to perform (long-term) msp, and the tasks of S-MEFC based classifier are: to dynamically set the FTs, to

Fig. 7. Enhanced multivariate degradation based modeling strategy

estimate the discrete states from msp and to stop the prediction process when the transition from degrading state to faulty state occurs (see Eq. 16). In comparison, the classical approach of multivariate prognostics (section I) does not take into account these important issues. The flow diagram of prognostics using proposed enhanced multivariate degradation modeling strategy and its dynamic threshold assignment procedure is shown in Fig. 7, where two main steps are highlighted: 1) off-line phase to learn univariate predictors and multivariate unsupervised classifiers, considering set of predictable features only, and 2) the on-line phase to simultaneously perform continuous state predictions and discrete state estimation, to dynamically assign FTs for RUL estimation. The detailed descriptions of each phase are given here after.

A. Off-line phase: learn predictors and classifiers

Given training dataset L_D (Eq. 11) having multidimensional features F_{Li} (or sensor measurements), that can be from multiple learning cases. According to dimension of feature space, $\ddot{n}$ univariate predictors P_i are build using SW-ELM algorithm and trained with data from $\ddot{m}$ cases (Fig. 8, left part). The features set is further reduced by post-treatment (section II-A1) to final set of predictable features according to learned predictors. This allows to retain or reject each potential

couple of "feature-model" for prognostics. Following that, an unsupervised classifier is build using S-MEFC algorithm for each learning case using multidimensional features F_{Li} that are predictable (Fig. 8, right part). Each classifier C_L can have different number of clusters (or states), which depends on a particular learning case. This enables to set FTs dynamically on-line.

B. On-line phase: perform predictions & discrete estimation

In the off-line phase, with predictable features $\ddot{n}$ univariate predictors and C_L unsupervised classifiers are build. For the on-line phase (Fig. 9), consider a new test case TF_i containing piece of trajectories from multidimensional signals (i.e., predictable ones) up to current time tc, from where prognostics should be initiated. *Prior* to RUL estimation task, the FT is assigned dynamically by looking at distance similarity (ED) among learned classifiers (from the off-line phase) and indexes of test data of case TF_i (e.g. from index= 1:tc). Note that, each index of TF can have similarity to a particular C_L. This problem can be simply resolved by applying "mode operation" to identify most frequently matched classifier with indexes of test data (TF). Suppose that, most of the indexes TF match classifier C_m, that is built in the off-line phase. Now, assume that the total number of clusters (or states) in C_m

Fig. 8. Off-line phase

Fig. 9. On-line phase

are **3**. This means 3rd state is the faulty one, and threshold for the test case TF_i is assigned according to that. When the matched C_m is identified, the severity of degradation at the current time tc can be also determined by distance similarity between tc and states of C_m. After assignment of FT and assessment of the severity of degradation, the RUL estimation task is initiated. Thus, with the proposed strategy, $\ddot{n}$ predictors perform msp (section II-A3), and simultaneously discrete states are estimated (by using C_m, see Fig. 6). The msp continues until their discrete states intersect FT (i.e., transition from degrading state to faulty, see Eq. 16).

IV. PHM CASE STUDY ON REAL DATA

A. Turbofan Engines datasets of IEEE PHM Challenge 2008

The aim of this case study is to validate the performances of the proposed enhanced multivariate degradation modeling. For this purpose, the data of turbofan engines are considered, which are available at NASA data repository [42]. Basically, the turbofan engines degradation simulations were carried out using the Commercial Modular Aero-Propulsion System Simulation test-bed developed by NASA for noisy sensor measurements. In order to train and test the prognostics model, we used text files "$train-FD001.txt$" composed of 100

training cases with different temporal lengths (Fig. 10), and "$test-FD001.txt$" also with 100 test cases. Each engine starts from a normal state but, due to some fault occurrence, begins to degrade. The degradation increases with time until failure takes place. Note that, the test data are composed of pieces of trajectories and their RUL is unknown. Each case either from the train or test file has 26 columns that represent: engine no., time in cycles, constant operational settings and sensor measurements (or time series features) contaminated with measurement noise. According to our previous work [12], only 8 features are considered from each case for the experiments i.e., sensor measurements 2, 3, 4, 8, 11, 13, 15, 17. As an example, consider the left part of Fig. 10, where run-to-failures data of "sensor 2" are shown from 100 engines in the training data. The data are highly noisy and should be filtered before feeding the prognostics model. The right part of Fig. 10 shows the life spans distribution of 100 engines in the training data. The variability in learning histories can affect the robustness of the prognostic model [25].

B. Performance evaluation and simulation setting

For results evaluation, estimated RULs are compared with actual RULs provided in the file "$rul-FD001.txt$". Most importantly, for a given test case TF, an interval I=[-10,13] is
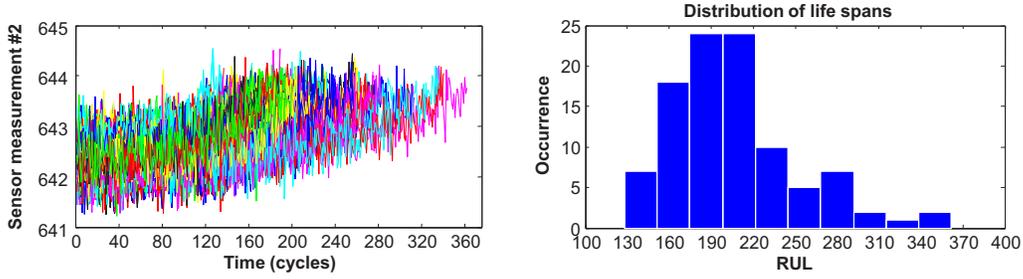
Fig. 10. Sensor measurement from 100 engines (training data) and life spans

considered to assess RUL estimates as on-time, early or late (Fig. 11). This interval is further used to compute final score for comparison (see [43]), which should be as low as possible. In PHM context, it is generally desirable to have early RUL estimates rather than late RULs, since the main aspect is to avoid failures. Therefore, for engines degradation scenario, an early RUL estimate is preferred over late RUL.

In brief, the scoring function for the challenge data is asymmetric around the true failure time to heavily penalize late RUL estimates as compared to early RULs [43]. However, with increase in error the scoring penalty grows exponentially. The scoring function (s) is given by Eq. 17 as follows:

$$s = \begin{cases} \sum_{i=1}^{\ddot{m}} e^{-(\frac{d}{a1})} - 1 \ for \ d < 0 \\ \\ \sum_{i=1}^{\ddot{m}} e^{(\frac{d}{a2})} - 1 \ for \ d \geq 0 \end{cases} \quad (17)$$

where, $a1$ and $a2$ are parameters to control the asymmetric preference, with values $a1 = 10$ and $a2 = 13$ (i.e., $I = [-10, 13]$). The RUL error is denoted by d (i.e., estimated RUL - Actual RUL), and $\ddot{m}$ represents the cases under test.
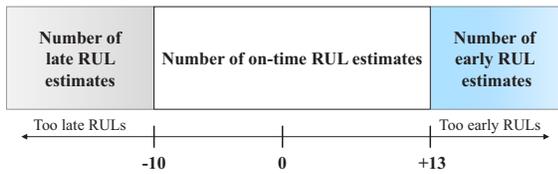
Fig. 11. RUL error interval [16]

Prognostics performances are further assessed by following criteria.

1) Total computation $time$ to learn and test entire dataset (i.e., fleet of 200 engines).
2) Accuracy of prognostics model evaluated by coefficient of determination ($R2$), that should be close to 1.

$$R2 = 1 - \frac{\sum_i (RUL_i - \widehat{RUL_i})^2}{\sum_i (RUL_i - \overline{RULs})^2} \quad (18)$$

where the numerator in the above equation measures the deviation of actual RUL_i from its predicted value $\widehat{RUL_i}$, and the denominator measures the deviation of actual RUL_i values from their mean $\overline{RULs}$.

According to proposed scheme (section II), SW-ELM is used to build univariate predictors (with respect to dimension of feature space). Assuming that a single model cannot guarantee accuracy of prediction, therefore for each feature 100 SW-ELM models (with different weights / bias) are learned and the model with minimum training error is selected. The network topology of each SW-ELM predictor is set as follows: 3 regressors input nodes, 5 hidden nodes, 1 output node and constant $C = 0.1$ (i.e., see [28] for details about parameter initialization). For the predictability metric (Eq. 1), the performance limit is set to $L = 0.5$. As for discrete state estimation (i.e., health assessment), the S-MEFC classifier is used with a neighborhood radius $ra = 0.4$, and the fuzziness parameter $\sigma = 0.38$.

C. Prognostics results on complete test data

1) Results comparison using all features & predictable features: according to previous discussions in section II-A1 prognostics should be performed only with predictable features, that leads to better RUL estimates. To avoid any confusion, the RUL estimates for turbofan data are expressed in units time i.e., number of cycles. In comparison to our previous work [12] from the experiments on training data, by learning 90 engines data and performing predictability analysis 10 engines data, we found that, among 8 features, only 5 are predictable, i.e., features {F1 ; F4 - F7} or sensor (2, 8, 11, 13, 15). Let us validate this assumption by estimating the RUL for each multidimensional test data set, by using, on one side, the whole set of pre-selected features {F1 - F8}, and on the other side, these final "predictability-based" selected features.

To estimate the RULs for both cases, SWELM and S-MEFC model are learned with 100 cases and tests are performed on 100 as well. It should be noted that for the learning and testing tasks, datasets are used with actual scales. As for example, the RUL estimation result with simultaneous prediction (by SWELM) and state estimation (by S-MEFC) involving pre-selected features {F1 - F8} from the first test case is shown in Fig. 12.

For RUL estimation, the dynamic FTs assignment is possible due to S-MEFC algorithm that can automatically determine different clusters (or states) among multidimensional data. Obviously, this is a realistic approach, because each machine can have different level of degradation from good state to faulty state, and any assumption about states should be avoided. For
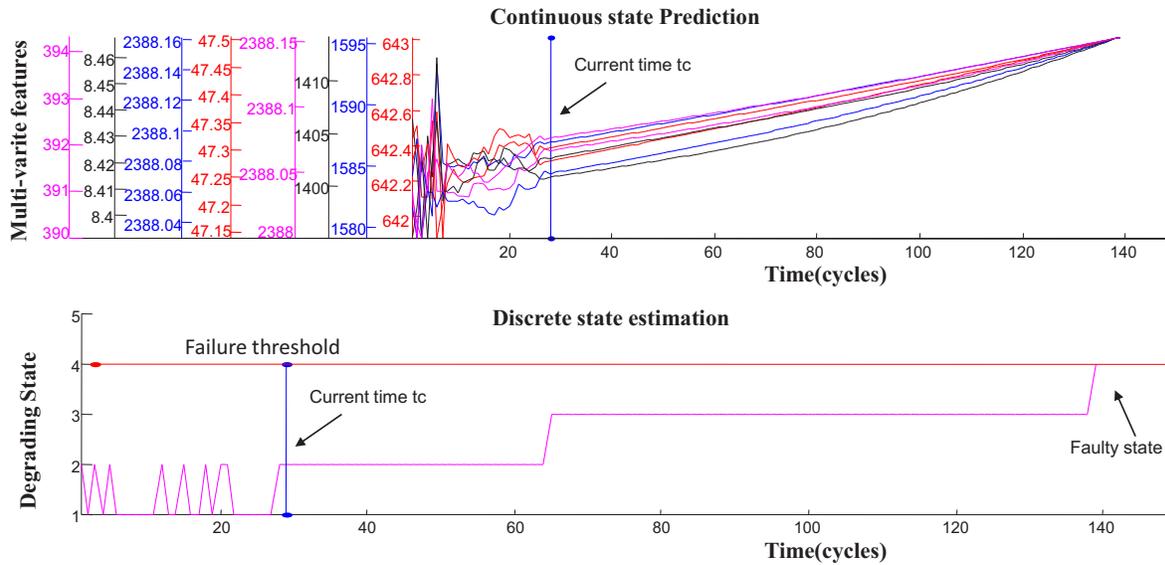
Fig. 12. RUL estimation with all features ({F1 - F8}) - Turbofan engine 1
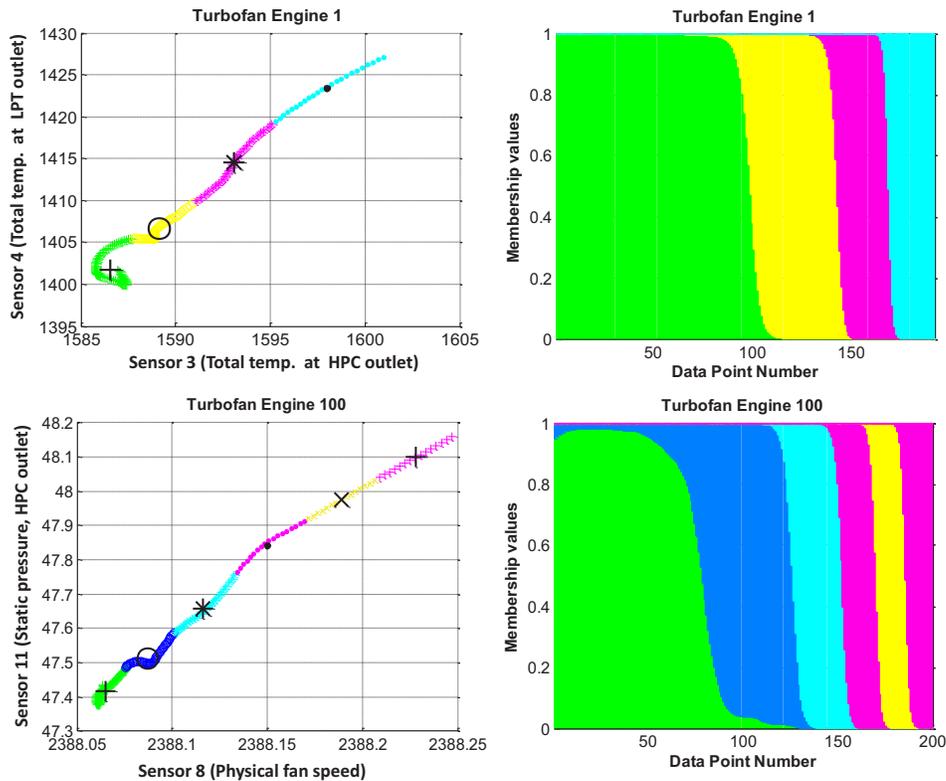
Fig. 13. Visualization of classes and data point membership functions

e.g. consider Fig. 13, where clustering results on run-to-failure data on engine 1 and 100, from file "$train - FD001.txt$" are provided for different sensor measurements.

The right part of the figure shows a color code sequence which corresponds to the memberships of data points in the clusters shown on the left part of Fig. 13 having the same color. In case of turbofan engine 1, data are automatically partitioned into 4 groups, whereas for turbofan engine 100 data are automatically

partitioned into 6 groups using S-MEFC algorithm. Similarly, each classifier C_L (for a particular engine) can represent different states of degradation.

The dynamic FTs are assigned by looking at distance similarity in multidimensional data of test case and learned classifiers C_L. Fig. 14, shows variations of FTs, while performing prognostics with all features {F1 - F8} and with predictable features {F1 ; F4 - F7} for complete test data of 100 engines.
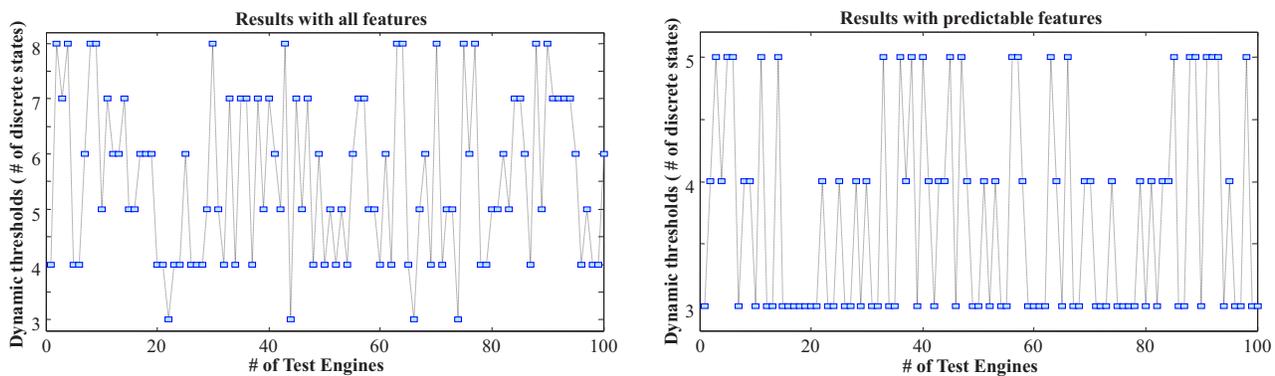
Fig. 14. Dynamic FTs assignment results
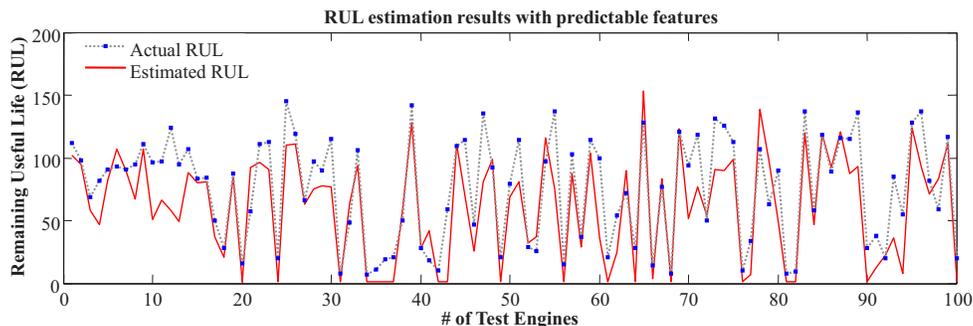
Fig. 15. Actual RUL vs. estimated RUL (100 Tests)

Mainly, the difference of the results is due to the number of features used to build a classifier C_L (because unpredictable features can lead to poor prognostics results).

Now, lets validate our proposed methodology to get correct RUL estimates falling in an interval $I = [-10, 13]$ (Fig. 11). Table. I summarizes performances for all tests initiated at current time tc, and compares prognostics results with all features {F1 - F8}, with predictable features {F1 ; F4 - F7} and classical multivariate degradation modeling approach [16].

TABLE I
PROGNOSTICS MODEL RESULTS COMPARISON FOR 100 TEST ENGINES

Criteria	All features	Predictable features	By [16]
RUL error pdf	I=[-85,74]	I=[-39,60]	I=[-85,120]
On-time RULs	32	48	53
Early RULs	34	40	36
Late RULs	34	12	11
$R2$	0.55	0.614	N/A
$Time$	5m 33sec	3m 54sec	N/A
Total score	4463	1046	N/A

Among 100 tests cases, with predictable features RUL estimates of **48** cases fall in interval I=[-10,13], i.e., on-time RUL estimates. The amount of early RUL estimates is **40**, and amount of late RUL estimates is only **12**. Note that early estimates are preferable than late estimates that can cause catastrophic situations [43].

The estimated RULs with the proposed approach are also compared with actual RULs provided in the file

"$rul - FD001.txt$", and the overall accuracy achieved is **R2=0.614** with a global score of **1046**, whereas with all features score is 4463. Most importantly, the RUL error distribution with the proposed approach (with predictable features) has lowest span **I=[-39,60]** as compared to other methods (Fig. 16). This shows robust performance of the prognostics model when exposed to tests without any knowledge about the RULs and their variability. Such behavior is depicted in Fig. 15, where the estimated RULs are closely mapped to actual RULs.

Besides that, it is also important for a prognostics model to meet real time constraints. In this context, the total computation time with the proposed approach to learn and test the data from the fleet of **200** turbofan engines is just **3m 54sec**.

2) Results comparison with other approaches: as expected, predictability based prognostics model has much better results as compared to prognostics with all features (including unpredictable ones). Obviously, features that are hard to be predicted can lead to poor prognostics results. But still, it is required to further compare the performances of the proposed "enhanced multivariate degradation modeling" strategy, to other approaches in the literature. Unfortunately the authors who use data "$train - FD001.txt$" and "$test - FD001.txt$" do not clearly mention errors of RUL estimates and scoring (for each test) which prevents benchmarking approaches. In literature only one publication was made with same data of 200 turbofan engines (and also the same Features {F1 - F8}) by applying "classical multivariate degradation modeling"
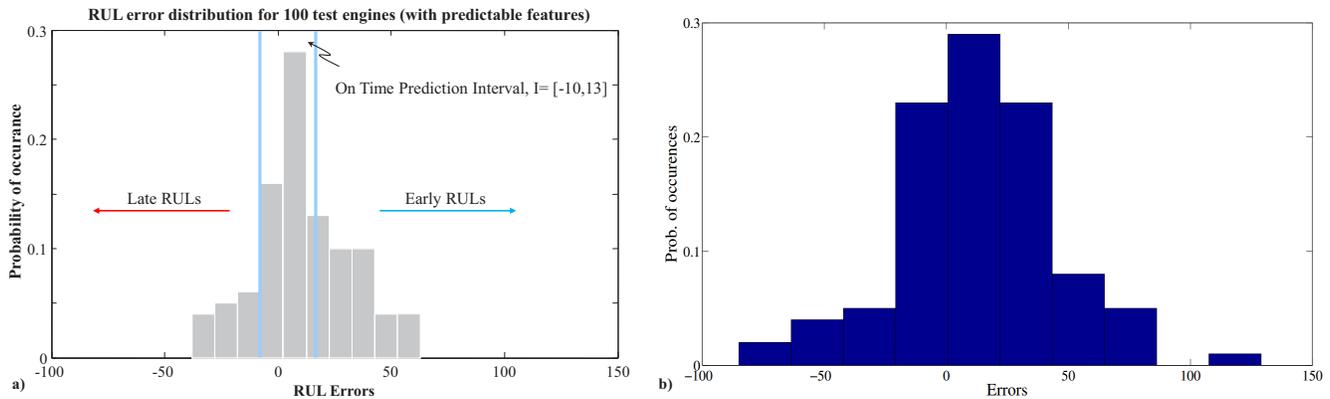
Fig. 16. RUL error distribution a) proposed approach and b) by [16]

strategy [16], but again limited information on results were given. Nevertheless, it will help to make a final comparison with RUL estimates achieved with our method.

The RUL error distribution obtained with proposed approach is compared with the recent results published by [16], (see Fig. 16). According to these results, the proposed approach has better precision of RUL estimates as indicated by the compact distribution of RUL errors (i.e., from **I=[-39, 60]**). In comparison, the results provided by [16] are less precise with a wider distribution of RUL errors (i.e., approximately from I=[-85, 120]). However, [16] achieved higher number of on-time estimates as compared to our method, i.e., **53** vs. 48. But, as the RUL error distribution has a wide spread on interval I=[-85, 120], this shows very early / late RUL estimates and will surely result in large score according to criteria of data challenge (Eq. 17). Relatively, when the proposed approach is evaluated on the basis of scoring criteria, the achieved total score is **1046**, (see Table. I). The score curve for all 100 tests is given in Fig. 17, where only **5%** RUL estimates are very early, which affects the scoring. However, the score plot clearly reflects that, the proposed method has preference for early RUL estimates rather than late RULs. Further, it is not possible to compare our score with other approaches in the literature, due to the absence of scoring results with this data (i.e., $train - FD001.txt$).
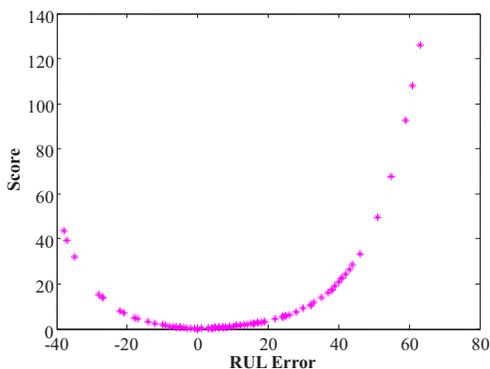
Fig. 17. Scores for 100 tests (enhanced multivariate degradation modeling)

Lastly, in case of the total computation time comparison, the proposed approach is much faster (i.e., **3m 54sec**) than "trajectory similarity based prediction approach" by [44], which could take few hours even with only data of **100** test cases (i.e., engines).

V. CONCLUSION

The goal of prognostics is to intelligently use condition monitoring data from an in-service machinery, and to estimate its RUL. Indeed, real machinery operates in highly non-linear environment and the monitored data could be of high variability, sensor noise, etc., which can impact on performances of prognostics approaches. In other words, complex and nonlinear nature of degradation phenomena can lead to poor prognostics results. In this paper, different strategies for data-driven prognostics are summarized and their issues are highlighted. Based on that, the developments focus on the proposition of predictability driven "enhanced multivariate degradation modeling" using two new machine learning approaches i.e., SW-ELM and S-MEFC. From application point of view and in comparison to existing data-driven prognostics approaches for RUL estimation, the proposed approach has following benefits.

- It works on actual scales of the data;
- It manages unlabeled data of different temporal lengths;
- It requires no assumption about the data and can dynamically estimate states of degrading machinery;
- It assigns failure thresholds dynamically without any *prior* knowledge;
- It requires only four parameters to be set by the user and computationally less expensive.

Results analysis on a case study of turbofan engines data (with different life spans) from PHM challenge 2008 show that, the use of multidimensional data (and predictable features) is beneficial in improving accuracy of RUL estimates. The future perspectives of this work are:

- to automate the initialization of two important parameters of the prognostic model, i.e., number of neurons in the hidden layer of SW-ELM, and the radius of neighborhood of a cluster in S-MEFC.

- to include data-pretreatment procedure and to integrate variable operating conditions as an inputs of the proposed prognostics model.

REFERENCES

[1] S. Uckun, K. Goebel, and P. Lucas, "Standardizing research methods for prognostics," in *Prognostics and Health Management International Conference on*, 2008, pp. 1–10.

[2] A. Saxena, J. Celaya, B. Saha, S. Saha, and K. Goebel, "Evaluating prognostics performance for algorithms incorporating uncertainty estimates," in *Aerospace Conference, 2010.* IEEE, 2010, pp. 1–11.

[3] S. A. Asmai, B. Hussin, and M. M. Yusof, "A framework of an intelligent maintenance prognosis tool," in *Computer Research and Development, 2010 Second International Conference on*. IEEE, 2010, pp. 241–245.

[4] E. Balaban and J. J. Alonso, "An approach to prognostic decision making in the aerospace domain," in *Annual Conference of the Prognostics and Health Management Society, Mineapolis, Sept. 23-27*, 2012.

[5] A. K. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical systems and signal processing*, vol. 20, no. 7, pp. 1483–1510, 2006.

[6] H. Li, D. Pan, and C. Chen, "Intelligent prognostics for battery health monitoring using the mean entropy and relevance vector machine," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 7, pp. 851–862, July 2014.

[7] A. Heng, S. Zhang, A. C. Tan, and J. Mathew, "Rotating machinery prognostics: State of the art, challenges and opportunities," *Mechanical Systems and Signal Processing*, vol. 23, no. 3, pp. 724–739, 2009.

[8] M. J. Daigle and K. Goebel, "Model-based prognostics with concurrent damage progression processes," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 3, pp. 535–546, 2013.

[9] J. Luo, K. R. Pattipati, L. Qiao, and S. Chigusa, "Model-based prognostic techniques applied to a suspension system," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 5, pp. 1156–1168, 2008.

[10] E. Zio and F. Di Maio, "A data-driven fuzzy approach for predicting the remaining useful life in dynamic failure scenarios of a nuclear system," *Reliability Engineering & System Safety*, vol. 95, no. 1, pp. 49–57, 2010.

[11] A. Mosallam, K. Medjaher, and N. Zerhouni, "Bayesian approach for remaining useful life prediction." *Chemical Engineering Transactions*, vol. 33, pp. 139–144, 2013.

[12] K. Javed, R. Gouriveau, and N. Zerhouni, "Novel failure prognostics approach with dynamic thresholds for machine degradation," in *IECON 2013-39th Annual Conference on IEEE Industrial Electronics Society, Vienna, Austria.* IEEE, pp. 4402–4407.

[13] O. E. Dragomir, R. Gouriveau, N. Zerhouni, F. Dragomir *et al.*, "Framework for a distributed and hybrid prognostic system." in *4th IFAC Conference on Management and Control of Production and Logistics, MCPL'2007.*, vol. 3, 2007, pp. 431–436.

[14] E. Ramasso and R. Gouriveau, "Prognostics in switching systems: Evidential markovian classification of real-time neuro-fuzzy predictions," in *Prognostics and Health Management Conference, 2010. PHM'10.* IEEE, 2010, pp. 1–10.

[15] E. Ramasso and R. Gouriveau, "Remaining useful life estimation by classification of predictions based on a neuro-fuzzy system and theory of belief functions," *IEEE Trans. Rel.*, vol. 63, no. 2, pp. 555–566, June 2014.

[16] E. Ramasso, M. Rombaut, and N. Zerhouni, "Joint prediction of continuous and discrete states in time-series based on belief functions," *IEEE Trans. Cybern.*, vol. 43, no. 1, pp. 37–50, 2013.

[17] L. Serir, E. Ramasso, and N. Zerhouni, "Time-sliced temporal evidential networks: The case of evidential hmm with application to dynamical system analysis," in *Prognostics and Health Management (PHM), 2011 IEEE Conference on*, June 2011, pp. 1–10.

[18] K. Javed, R. Gouriveau, R. Zemouri, N. Zerhouni *et al.*, "Features selection procedure for prognostics: An approach based on predictability," *8th IFAC Int. Symp. On Fault Dectection, Supervision and Safety of Technical Processes.*, pp. 25–30, 2012.

[19] R. Zemouri, R. Gouriveau, and N. Zerhouni, "Improving the prediction accuracy of recurrent neural network by a pid controller," *International Journal of Systems Applications, Engineering & Development*, vol. 4, no. 2, pp. 19–34, 2010.

[20] C.M. Bishop, *Neural Networks for Pattern Recognition.* Oxford University Press, Walton Street, Oxford OX2 6DP, 1995.

[21] H. Jaeger, *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the echo state network approach.* GMD-Forschungszentrum Informationstechnik, 2002.

[22] R. Singh and S. Balasundaram, "Application of extreme learning machine method for time series analysis," *International Journal of Intelligent Technology*, vol. 2, no. 4, pp. 256–262, 2007.

[23] M. R. Meireles, P. E. Almeida, and M. G. Simões, "A comprehensive review for industrial applicability of artificial neural networks," *IEEE Trans. Ind. Electron.*, vol. 50, no. 3, pp. 585–601, 2003.

[24] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: A new learning scheme of feedforward neural networks," in *International Joint Conference on Neural Networks*, Budapest, Hungary, 2004.

[25] K. Javed, R. Gouriveau, N. Zerhouni, R. Zemouri, and X. Li, "Robust, reliable and applicable tool wear monitoring and prognostic: approach based on an improved-extreme learning machine," in *IEEE Conference on Prognostics and Health Management*, Denver, CO, USA, 2012.

[26] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Cybern.*, vol. 42, no. 2, pp. 513–529, 2012.

[27] J. Luo, C.-M. Vong, and P.-K. Wong, "Sparse bayesian extreme learning machine for multi-classification," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 25, no. 4, pp. 836–843, April 2014.

[28] K. Javed, R. Gouriveau, and N. Zerhouni, "SW-ELM: A summation wavelet extreme learning machine algorithm with *a priori* parameter initialization," *Neurocomputing*, vol. 123, pp. 299–307, 2014.

[29] A. Banakar and M. F. Azeem, "Artificial wavelet neural network and its application in neuro-fuzzy models," *Applied Soft Computing*, vol. 8, no. 4, pp. 1463–1485, 2008.

[30] Y. Oussar and G. Dreyfus, "Initialization by selection for wavelet network training," *Neurocomputing*, vol. 34, no. 1, pp. 131–143, 2000.

[31] D. Nguyen and B. Widrow, "Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights," in *Neural Networks, 1990., 1990 IJCNN International Joint Conference on.* IEEE, 1990, pp. 21–26.

[32] C. R. Rao and S. K. Mitra, *Generalized Inverse of Matrices and its Applications.* John Wiley and Sons, New York, 1971.

[33] R. Gouriveau and N. Zerhouni, "Connexionist-systems-based long term prediction approaches for prognostics," *IEEE Trans. Rel.*, vol. 61, no. 4, pp. 909–920, 2012.

[34] M. El-Koujok, R. Gouriveau, and N. Zerhouni, "Reducing arbitrary choices in model building for prognostics: An approach by applying parsimony principle on an evolving neuro-fuzzy system," *Microelectronics Reliability*, vol. 51, pp. 310–320, 2011.

[35] T. Warren Liao, "Clustering of time series dataŮa survey," *Pattern Recognition*, vol. 38, no. 11, pp. 1857–1874, 2005.

[36] S. L. Chiu, "Fuzzy model identification based on cluster estimation," *Journal of intelligent and Fuzzy systems*, vol. 2, no. 3, pp. 267–278, 1994.

[37] R.-P. Li and M. Mukaidono, "A maximum-entropy approach to fuzzy clustering," in *Fuzzy Systems, 1995. International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium., Proceedings of 1995 IEEE International Conference on*, vol. 4. IEEE, 1995, pp. 2227–2232.

[38] C. Doan, S. Liong, and D. Karunasinghe, "Derivation of effective and efficient data set with subtractive clustering method and genetic algorithm." *Journal of Hydroinformatics*, vol. 7, pp. 219–233, 2005.

[39] K. Bataineh, M. Naji, and M. Saqer, "A comparison study between various fuzzy clustering algorithms," *Editorial Board*, vol. 5, no. 4, p. 335, 2011.

[40] B. H. Nystad, G. Gola, and J. E. Hulsund, "Lifetime models for remaining useful life estimation with randomly distributed failure thresholds," in *First european conference of the prognostics and health management society*, vol. 3, 2012.

[41] S. Cheng and M. Pecht, "A fusion prognostics method for remaining useful life prediction of electronic products," in *Automation Science and Engineering, CASE 2009. IEEE International Conference on.* IEEE, 2009, pp. 102–107.

[42] NASA, "Prognostic data repository http://ti.arc.nasa.gov/tech/dash/pcoe/prognostic-data-repository/."

[43] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *International Conference on Prognostics and Health Management, 2008.* IEEE, 2008, pp. 1–9.

[44] T. Wang, "Trajectory similarity based prediction for remaining useful lifeestimation," Ph.D. dissertation, University of Cincinnati, 2010.