Prognostics With Variational Autoencoder by Generative Adversarial Learning

Yu Huang, *Student Member, IEEE*, Yufei Tang, *Member, IEEE*, and James VanZwieten

***Abstract*—Prognostics predicts the future performance progression and remaining useful life (RUL) of in-service systems based on historical and contemporary data. One of the challenges in prognostics is the development of methods that are capable of handling real-world uncertainties that typically lead to inaccurate predictions. To alleviate the impacts of uncertainties and to achieve accurate degradation trajectory and RUL predictions, a novel sequence-to-sequence predictive model is proposed based on a variational autoencoder that is trained with generative adversarial networks. A long short-term memory network and a Gaussian mixture model are utilized as building blocks so that the model is capable of providing probabilistic predictions. Correlative and monotonic metrics are applied to identify sensitive features in the degradation progress, in order to reduce the uncertainty induced from raw data. Then, the selected features are concatenated with one-hot health state indicators as training data for the model to learn end of life without the need for prior knowledge of failure thresholds. Performance of the proposed model is validated by health monitoring data collected from real-world aeroengines, wind turbines, and lithium-ion batteries. The results demonstrate that significant performance improvement can be achieved in long-term degradation progress and RUL prediction tasks.**

***Index Terms*—Gaussian mixture model (GMM), generative adversarial learning, long short-term memory (LSTM), prognostics and health management (PHM), remaining useful life (RUL), variational autoencoder (VAE).**

Nomenclature

Acronyms

EoL	End of life.
GAN	Generative adversarial networks.
GMM	Gaussian mixture model.
LSTM	Long short-term memory.
MCM	Machine condition monitoring.
PHM	Prognostics and health management.
RNN	Recurrent neural networks.
RUL	Remaining useful life.
VAE	Variational autoencoder.

I. Introduction

PROCESS safety, system reliability, and product quality are becoming increasingly essential in the modern industry [1]. Machine condition monitoring (MCM), a maintenance strategy that involves the repair and replacement of damaged parts to reduce the total life cycle costs, is a vital part of many industries, such as aerospace, energy, automotive, and heavy industry. Traditional strategies, such as corrective (breakdown) and preventive (scheduled) maintenance, are becoming less capable of meeting the increasing industrial demand for efficiency and reliability [2]. Prognostics and health management (PHM) is a novel paradigm that enables real-time health assessment and future condition prediction. PHM incorporates various disciplines (e.g., sensing technologies, signal processing, machine learning, and reliability analysis) and provides an intelligent MCM strategy to maintain a system's originally intended functions [3] or even distinguish whether a local malfunction will affect the key-performance-indicators of the whole system [1].

Prognostics of in-service systems is a pillar of PHM that can be sorted into two types: 1) remaining useful life (RUL) evaluation (i.e., event prediction); and 2) future degradation estimation (i.e., event progression prediction). Data-driven approaches, which use the information of current and previous usage conditions to identify the characteristics of the contemporary degradation state and to predict the future trajectory, have been regarded as a powerful solution for prognostics [4] and achieve success in cyber-physical systems [5], [6]. Machine learning, as the most common data-driven technique, is able to act as a bridge connecting big machinery data and intelligent prognostics [5]. For example, Elforjani *et al.* [7] employed three supervised machine learning techniques: support vector machine, Gaussian process regression, and multilayer neural network to estimate RUL for slow-speed naturally degrading bearings using acoustic technology. Furthermore, a deep convolutional neural network has been proposed to map monitored feature data to machine health status [8]. The combination of neural networks and

Manuscript received March 15, 2020; revised July 19, 2020, October 25, 2020, and December 17, 2020; accepted January 12, 2021. Date of publication January 28, 2021; date of current version September 29, 2021. This work was supported in part by the U.S. National Science Foundation through Grant ECCS-1809164 and Grant OAC-2017597 and in part by the U.S. Department of Energy, Water Power Technologies Office, under Award DE-EE0008955. *(Corresponding author: Yufei Tang.)*

Yu Huang and Yufei Tang are with the Department of Computer & Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL 33431 USA (e-mail: yhwang2018@fau.edu; tangy@fau.edu).

James VanZwieten is with the Department of Civil, Environmental, and Geomatics Engineering, Florida Atlantic University, Boca Raton, FL 33431 USA (e-mail: jvanzwi@fau.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIE.2021.3053882.

Digital Object Identifier 10.1109/TIE.2021.3053882

fuzzy systems has been employed successfully to capture more information for PHM [9]. Recently, a so-called "vanilla" long short-term memory (LSTM) network has been utilized in [10] to improve the accuracy of RUL prediction for complicated industrial processes.

However, the above-stated data-driven-based prognostic systems have the following potential concerns.

1) Feature representation: The data for prognostics are usually formulated sequentially, where hidden features behind these sequences are vital for representing a system's health condition. However, handcrafted features may not perfectly represent degradation throughout the lifetime.
2) Prior domain knowledge: RUL is calculated by subtracting end of life (EoL) from a system's current cycle. Generally, EoL is predicted: a) by mapping features to piecewise linear RUL curves [11] or using the linear relation between features and EoLs (e.g., Max.E-EoL [12]); b) when the degradation level reaches single or multiple predefined thresholds [13] (e.g., E-trend [12]). Both methods require sufficient expert knowledge either to build linear relations or to define thresholds that differ between scenarios, consequently hindering their flexibility.
3) Multimode degradation: In reality, a system can degrade in different manners (i.e., varied degradation mechanisms) even though it undergoes the same operation. Most prior or current prognostic models have been conducted on simple, naturally degenerated data with fixed initial parameters. Therefore, it is critical to make the predictive model adaptive to different degradation modes.

Various machine-learning algorithms have been applied in prognostics; however, it was demonstrated in [14] that feature representation determines the upper-bound performance of models. In many cases, degradation data cannot be collected directly, with system feedback used as an alternative. For instance, bearing vibration signals [15], [16] are commonly used for gearbox RUL prediction. It is difficult to extract effective features enriched with degradation characteristics from measurements directly, and conventional signal processing and feature extraction techniques often limit the ability to identify intricate correlations [17]. Therefore, the instrumentation and feature extraction scheme should be carefully developed. A complete data-driven method has been proposed in [18] to automatically produce system health indicators, without *a priori* knowledge of system monitoring or signal processing. A local feature-based gated recurrent unit network has been proposed in [19] to generate feature sequences without requiring high-level expert knowledge.

RUL prediction is achieved by subtracting the current cycle from the predicted EoL. The most commonly used method for predicting EoL includes labeling the training data auxiliary, where each sample is required to associate with its RUL label as a target. In this case, the piecewise linear method [11] is usually adopted. This requires extra work and is generally very time-consuming. Moreover, if the available label information is limited, the advantage of machine learning could be minimal. To overcome this, generative adversarial networks (GANs)-based models were proposed in [20] and [21] to cope with the insufficiency of health data for asset reliability prediction. Another method to predict EoL requires defining a failure threshold in advance. EoL is, therefore, assumed to occur when a health indicator exceeds that threshold. For example, an appropriate threshold is required to separate the hyperplane of the high-dimensional features in support-vector machines. However, sufficient expert knowledge of critical components' failure thresholds is not always readily available and human factors introduce much uncertainty, which is difficult to model and brings complexity to the analysis and synthesis procedures [5]. Furthermore, it is not appropriate to use a single threshold to summarize all failure modes.

The degradation progressions for the same mechanical system are variant. It may undergo multimode degradation triggered by many inducements, including enclosure problems, excessive operation, lack of maintenance, and corrosive environments. Mogren [22] suggests that GANs [23] are a viable way of modeling a distribution over different types of sequential data. Ha *et al.* [24] demonstrate that Gaussian mixture model (GMM)-recurrent neural networks (RNNs) can learn to forecast using an immense amount of observations from multiscenarios.

Inspired by previous work, this article proposes a novel data-driven approach based on GAN, focusing on enhancing the predictions of long-term degradation and RUL without predefining specific component failure thresholds. The generator in GAN is a novel adaptation of sequence-to-sequence variational autoencoder (VAE) derived from the combination of LSTM and GMM and the discriminator is a bidirectional LSTM. The contribution and intellectual merit of this research are twofold.

1) An LSTM-GMM-based VAE as a generative model is proposed, which is fed with time-series data and combined with a one-hot health indicator to bypass low-accuracy prediction generated from an imprecise predefined failure threshold. In this model, the concatenation of GMM with LSTM networks allows modeling and predicting of different modes of degradation scenarios in a single neural network conditioned on previous records.
2) The proposed VAE model is trained through an adversarial learning approach, i.e., GAN, to enhance the accuracy and robustness of long-term degradation and RUL prediction. The model's effectiveness is quantified by health data collected from various real-world engineering systems, including aeroengines, wind turbines, and lithium-ion batteries.

The remainder of this article is arranged as follows. Section II formulates the problem. Section III presents the methodology in detail. Section IV presents the experimental results. Finally, Section V concludes this article.

II. Problem Statement

Let $\boldsymbol{x}^{(i)}$ denote a vector of multivariate sensor measurements such that $\boldsymbol{x}^{(i)} = [x^{(i;1)}, \ldots, x^{(i;m)}]$, where m is the number of sensors. Formally, a sensor measurement sequence is described

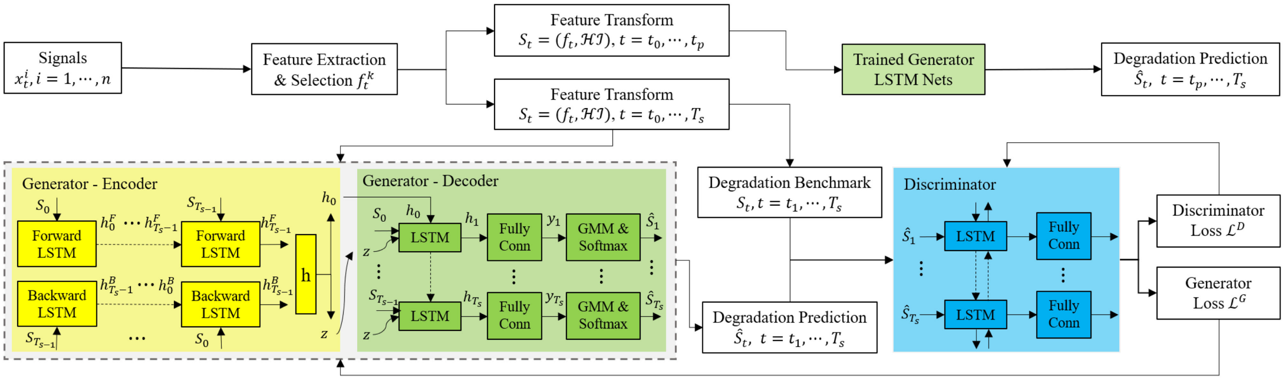

Fig. 1. Schematic diagram of the VAE-GAN. The generator is a sequence-to-sequence VAE, consisting of a “bidirectional” LSTM-based encoder, and an autoregressive LSTM-based decoder. The discriminator is a bidirectional LSTM.

by $\mathcal{X} = \{\boldsymbol{x}^{(0)}, \ldots, \boldsymbol{x}^{(n-1)}\}$, where $\boldsymbol{x}^{(i)} \subseteq \mathcal{R}^m$ and n is the total time steps of observations. The ground-truth of prediction at the next time step is denoted by $\mathcal{Y} = \{\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(n)}\}$, where $\boldsymbol{y}^{(i)} = \boldsymbol{x}^{(i)}$, $i = 1, \ldots, n$. The dataset $\mathcal{D}$ is defined by $\mathcal{D} = \{(\boldsymbol{x}^{(i-1)}, \boldsymbol{y}^{(i)})\}_{i=1,\ldots,n}$. In general, the data-driven prognostics approach is to learn the best predictor of run-to-failure degradation from the previously observed data, i.e., a training dataset $\mathcal{D}^T$. Then, based on the prediction from the trained model, RUL can be estimated at each time step. This problem can be formulated by finding a nonlinear mapping function $\mathcal{F} : \boldsymbol{x}' \rightarrow \boldsymbol{z}$, with a latent variable $\boldsymbol{z} \subseteq \mathcal{R}^{N_z}$ and $N_z < m$. Subsequently, the optimal predictor can be formulated as the function of $\boldsymbol{z}$ shown as follows:

$$f_{\gamma}(\boldsymbol{z}) = \arg\max_{\gamma} p(\boldsymbol{y} \,|\boldsymbol{z}, \gamma) \tag{1}$$

where γ is the parameter of the nonlinear mapping function that needs to be optimized through training $f_{\gamma}(\boldsymbol{z})$. The primary goal of this article is to develop a data-driven approach to learn the nonlinear mapping function f_{γ} for degradation progress modeling and stable RUL prediction.

III. Methodology

A. Feature Extraction and Selection

As shown in Fig. 1, feature extraction and selection is an essential first-phase preparation of prognostics. In this process, critical features that contain sufficient degradation signatures will be identified to increase the efficiency and reliability of prognostics by 1) reducing the cost of feature measurement and 2) minimizing the dimensions of data required to describe the degradation progress [17], [25].

The majority of mechanical systems normally undergo gradual degradation rather than breaking down unexpectedly. In consideration of the reality that the ideal features for prognosis should practically indicate the system degradation trajectory throughout the lifetime, i.e., should be monotonous, Pearson’s correlation [26] and monotonic metrics [27] are implemented to evaluate the degradation-sensitivity of measurements $\boldsymbol{x}$ from the initially sampled raw data $\mathcal{X}$. Specifically, the monotonic metric in (2) evaluates the ascending/descending trend of features, and the Pearson’s correlation metric in (3) assesses the correspondence between features and time, shown in the following:

$$\text{mono} = \left| \frac{dx_t^m > 0}{T-1} - \frac{dx_t^m < 0}{T-1} \right| \tag{2}$$

$$\text{cor} = \frac{\left| \sum_{t=1}^{T} \left(x_t^m - \overline{x}^m\right)\left(t - \overline{t}\right) \right|}{\sqrt{\sum_{t=1}^{T} \left(x_t^m - \overline{x}^m\right)^2 \sum_{t=1}^{T} \left(t - \overline{t}\right)^2}} \tag{3}$$

where $\boldsymbol{x}$ could be sensor measurements (e.g., vibration) or statistics (e.g., amplitude in frequency spectrum), x_t^m is the mth sensor/statistic at time step t of sequence length T, $\overline{(\cdot)}$ and $d(\cdot)$ denotes the mean and differential operation, respectively.

Feature selection is accomplished based on a linear combination of correlative and monotonic performance, i.e., $\delta \cdot \text{mono} + (1 - \delta) \cdot \text{cor}$, where δ is a tradeoff hyperparameter set to 0.5 here. A sensor/statistic with the k highest criteria value will be selected as the most degradation-sensitive features $\boldsymbol{f} \subseteq \mathcal{R}^k$ to discard irrelevant or redundant ones.

B. Feature Transformation

Predominantly, to solve the nonlinear equation $y_{\text{thold}} = \mathcal{F}(t_{\text{EoL}}; \hat{\boldsymbol{\theta}})$ for RUL prediction, it is necessary to find the time cycle (t_{EoL}) when the degradation reaches a certain level y_{thold}. Therefore, the RUL can be determined from $\text{RUL} = t_{\text{EoL}} - t_{\text{current}}$. Defining y_{thold} requires sufficient expert knowledge of the system. When the system is intricate and the failure modes are various, it is unmanageable to define a certain threshold to represent all failures.

With the intention of enabling the model learning of t_{EoL}, the selected feature is concatenated with an additional health indicator ($\mathcal{HI}$) to represent the machinery degradation process instead of using it directly as follows:

$$\boldsymbol{S}_t = (\boldsymbol{f}_t, \mathcal{HI}) \tag{4}$$

where $\boldsymbol{f}_t \subseteq \mathcal{R}^k$, k is the dimension of selected features. Unlike the prognostic method in [28] that maps $\mathcal{HI}$s with RULs, here $\mathcal{HI} = (h_1, h_2)$ is generated by one-hot encoding with only two values. Specifically, (1,0) suggests that the equipment is healthy and currently in operation, whereas (0,1) suggests that the equipment has undergone failure and requires maintenance. The initial value is set as $\boldsymbol{S}_0 = (\mathbf{0}, 1, 0)$.

To develop a simple, robust approach that works well for a broad class of degradation series, the method first reformats each series to a fixed length of $T_{\max}$, where $T_{\max}$ is the longest sequence in the training dataset indicating the slowest degradation mode. In principle, $T_{\max}$ can be considered as a variable reading from the training data directly. For those sequence $\boldsymbol{S}$ whose length T_s is shorter than $T_{\max}$, $\boldsymbol{S}_t$ is set as $(\mathbf{0}, 0, 1)$ for $T_s \leq t < T_{\max}$ to make all series of the same length $T_{\max}$. In addition, min–max normalization and mean filtering are implemented in advance to reduce the influence of noise. The model training will be discussed in detail in the next section.

C. LSTM-GMM-Based VAE With Adversarial Training

The LSTM neural network is a significant branch of RNNs that are often used to model sequences of data [29]. However, a standard LSTM generates sequences with one data point at a time, which does not work for an explicit global sequence representation. Considering our problem scenario is similar to that in [20] and [30], VAE is a good option since it has been verified to be an efficient stochastic variational inference and learning algorithm that scales to large datasets. Moreover, VAE even works in the intractable case under some mild differentiability conditions [30]. Here, an LSTM- and GMM-based sequence-to-sequence VAE with adversarial training is proposed, referred to as VAE-GAN, seeking to incorporate distributed latent representations of the entire sequences (life-long degradation) with various degradation modes. The adversaries, a generator G and a discriminator D, are two different deep RNNs.

Generator: As shown in Fig. 1, the encoder in G consists of two LSTMs, taking the input in both directions to obtain two hidden states. Specifically, at each time step t, the encoder in the generator takes all available observations $\boldsymbol{S}_{T_s}^F$ as well as the same observations in opposite order $\boldsymbol{S}_{T_s}^B$ as inputs, and outputs two hidden states $\boldsymbol{h}_{T_s}^F$ and $\boldsymbol{h}_{T_s}^B$ as

$$\boldsymbol{h}_{T_s}^F = \mathcal{LSTM}\left(\boldsymbol{S}_{T_s}^F\right), \quad \boldsymbol{h}_{T_s}^B = \mathcal{LSTM}\left(\boldsymbol{S}_{T_s}^B\right). \tag{5}$$

Then, a fully connected layer is employed to map the concatenated hidden state $[\boldsymbol{h}_{T_s}^F; \boldsymbol{h}_{T_s}^B]$ into $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$. In addition, due to the nonnegativity of standard deviation, the exp operation is applied to $\boldsymbol{\sigma}$ as

$$\boldsymbol{\mu} = W_\mu[\boldsymbol{h}_{T_s}^F; \boldsymbol{h}_{T_s}^B] + b_\mu \tag{6}$$

$$\boldsymbol{\sigma} = W_\sigma[\boldsymbol{h}_{T_s}^F; \boldsymbol{h}_{T_s}^B] + b_\sigma, \quad \hat{\boldsymbol{\sigma}} = \exp\left(\frac{\boldsymbol{\sigma}}{2}\right) \tag{7}$$

and then, the latent vector $\boldsymbol{z}$ is set up as follows [30]:

$$\boldsymbol{z} = \boldsymbol{\mu} + \hat{\boldsymbol{\sigma}} \odot \epsilon \tag{8}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$ and $\odot$ signifies the elementwise product. Therefore, the latent vector $\boldsymbol{z}$, illustrated in Fig. 2, as an example, is a learned vector of dimension N_z constrained on the input sequence instead of a deterministic yield given certain input. Such an encoding scheme enables modeling of the case under mild different conditions [30].

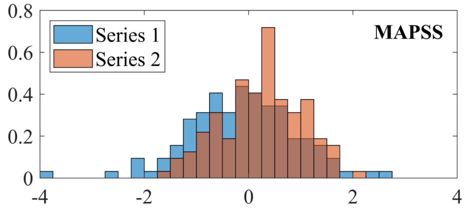

Fig. 2. Illustration of the encoded $\boldsymbol{z}$ of two random series of MAPSS dataset.

The decoder in G is a unidirectional LSTM network. At each time step t, the decoder takes in the previous data S_{t-1} and the latent vector $\boldsymbol{z}$ as a concatenated input $\boldsymbol{x}_{t-1}$. This input format implies that the generated consequent hidden state is conditioned on the latent $\boldsymbol{z}$ sampled from the encoder that is trained end-to-end along the decoder. The computation of the decoder can be described as follows:

$$\boldsymbol{h}_t = \mathcal{LSTM}\left(\boldsymbol{x}_{t-1}, \boldsymbol{h}_{t-1}\right) \tag{9}$$

where the inputs of the model are $\boldsymbol{x}_{t-1}$ and $\boldsymbol{h}_{t-1}$, and the initial hidden states h_0 of the decoder are the yield of a connected layer, that is, $h_0 = \tanh(W_z z + b_z)$. The outputs at each time step are the hidden states $\boldsymbol{h}_t$, which are the parameters for a probability distribution of the consequent data $\boldsymbol{S}_t$.

A fully connected layer is used to project the hidden state $\boldsymbol{h}_t$ into the output $\boldsymbol{y}_t$, $\boldsymbol{y_t} \subseteq \mathcal{R}^{3M+2}$, which can be split into M mixed Gaussian distributions to describe f_t and one categorical (p_1, p_2) distribution to describe health indicator $\mathcal{HI}$

$$\begin{aligned}\boldsymbol{y}_t &= \boldsymbol{W}_y \boldsymbol{h}_t + \boldsymbol{b}_y \\ &= [(\hat{\pi}_1, \mu_1, \hat{\sigma}_1), \ldots, (\hat{\pi}_M, \mu_M, \hat{\sigma}_M), (\hat{p}_1, \hat{p}_2)].\end{aligned} \tag{10}$$

The feature f_t in $\boldsymbol{S}_t$ described by the GMM with M normal distributions at each time step is given as

$$p(f_t) = \sum_{i=1}^{M} \pi_i \mathcal{N}(f_t | \mu_i, \sigma_i) \tag{11}$$

where μ_i and σ_i are the mean and standard deviation of the ith univariate normal distribution, respectively; $\boldsymbol{\pi}$ is a categorical distribution of length M with $\sum_{i=1}^{M} \pi_i = 1$, representing the mixture weights of the GMM.

Due to the probability constraint and the nonnegativity of standard deviations, exp and softmax operations are employed. The probabilities for the categorical distributions are calculated using the outputs as logit values

$$\sigma_i = \exp(\hat{\sigma}_i) \tag{12}$$

$$\pi_k = \frac{\exp(\hat{\pi}_k)}{\sum_{i=1}^{M} \exp(\hat{\pi}_i)}, k = 1, 2, \ldots, M \tag{13}$$

$$p_k = \frac{\exp(\hat{p_k})}{\sum_{i=1}^{2} \exp(\hat{p_i})}, k = 1, 2. \tag{14}$$

Discriminator: The discriminator D is a bidirectional LSTM network that allows us to take into account the input time series in both directions [22]. The outputs of each LSTM cell in D are fed into a fully connected layer with weights shared across time steps. One sigmoid output per cell is then averaged to the final decision for the sequence.

Loss and Training: The model is trained by simultaneously updating the discriminative distribution so that it discriminates between samples from the data generating distribution $p_{\text{data}}(\boldsymbol{S})$ and from those of the generative distribution $p_g(\hat{\boldsymbol{S}})$[23]. First, the optimization of the discriminator D given generator G is described as follows. Similar to the training of Sigmoid-function-based classifiers, it involves minimizing the cross entropy. The discriminator loss function $\mathcal{L}^D$ is formulated as follows:

$$\mathcal{L}^D(\boldsymbol{\theta}_d, \boldsymbol{\theta}_g) = \mathcal{L}^D_{\text{GAN}}(\boldsymbol{\theta}_d, \boldsymbol{\theta}_g) + \alpha L^D_2(\boldsymbol{\theta}_d) \tag{15}$$

where $\mathcal{L}^D_{\text{GAN}}(\boldsymbol{\theta}_d, \boldsymbol{\theta}_g)$ is the standard GAN loss and $L^D_2(\boldsymbol{\theta}_d)$ is the standard L_2 regularization defined as follows:

$$\mathcal{L}^D_{\text{GAN}} = \frac{1}{T_{\max}} \sum_{t=1}^{T_{\max}} \left[\log D(\boldsymbol{S}_t) + \log\left(1 - D(\hat{\boldsymbol{S}}_t)\right)\right] \tag{16}$$

$$\mathcal{L}^D_2 = \|\boldsymbol{\theta}_d\|_2 \tag{17}$$

where $\boldsymbol{S}_t$ is sampled from the ground truth degradation data and $G(\boldsymbol{x}_t) = \hat{\boldsymbol{S}}_t$ is the corresponding generated samples.

Then, fix D and optimize G to minimize the discrimination accuracy of D. The reconstructed loss function (18) is the sum of four terms: the standard GAN loss in (19) of generator $\mathcal{L}^G_{\text{GAN}}$, the log loss in (20) of feature variation $\mathcal{L}^G_f$, the log loss in (21) of health state terms $\mathcal{HI}$, and the Kullback–Leibler divergence loss $\mathcal{L}^G_k$ in (22) that representing the difference between the distribution of latent vector $\boldsymbol{z}$ with a Gaussian distribution with zero mean and unit variance

$$\mathcal{L}^G = \mathcal{L}^G_{\text{GAN}} + \mathcal{L}^G_f + \mathcal{L}^G_h + \mathcal{L}^G_k \tag{18}$$

$$\mathcal{L}^G_{\text{GAN}}(\boldsymbol{\theta}_d, \boldsymbol{\theta}_g) = \frac{1}{T_{\max}} \sum_{t=1}^{T_{\max}} \left[\log\left(1 - D(G(\boldsymbol{x}_t))\right)\right] \tag{19}$$

$$\mathcal{L}^G_f = -\frac{1}{T_{\max}} \sum_{t=1}^{T_s} \log\left(\sum_{k=1}^{M} \pi_{i,k} \mathcal{N}\left(f_t | \mu_{t,k}, \sigma_{t,k}\right)\right) \tag{20}$$

$$\mathcal{L}^G_h = -\frac{1}{T_{\max}} \sum_{t=1}^{T_{\max}} \left[h^t_1 \log\left(p^t_1\right) + h^t_2 \log\left(p^t_2\right)\right] \tag{21}$$

$$\mathcal{L}^G_k = \frac{1}{N_z}\left(1 + \boldsymbol{\sigma} - \boldsymbol{\mu}^2 - \exp(\boldsymbol{\sigma})\right). \tag{22}$$

Note that the GMM parameters modeling f_t beyond T_s are discarded when calculating $\mathcal{L}^G_f$, whereas $\mathcal{L}^G_h$ is calculated using all of the categorical distribution parameters modeling the health indicator $\mathcal{HI}$ until $T_{\max}$. Both terms are normalized by the total sequence length $T_{\max}$. The practice of loss definition $\mathcal{L}^G_h + \mathcal{L}^G_f$ was found to be more robust and empowers the VAE to learn the EoL in a straightforward manner. We empirically update the parameters of D for k times and then update G once. The optimization process between G and D alternates and improves their performance gradually. The global optimal solution is achieved if $p_{\text{data}} = p_g$, meaning when the discrimination ability of D has been improved to a high limit but cannot correctly discriminate further, it is thought that G has captured the distribution of the real data.

Algorithm 1: Present-to-EoL Degradation Prediction.

```
Input: feature series $\boldsymbol{S} = \boldsymbol{S}_t,\ t = 1, 2, \cdots, t_p$
initialize $h_0 = 0$, $\boldsymbol{S}_0 = (f_0, \mathcal{HI}) = (\mathbf{0}, 1, 0)$, $t = 0$;
Obtain $\boldsymbol{z}$ by encoding $\boldsymbol{S}$;
while $\mathcal{HI}\ ! = (0, 1)$ do
    Generate $\boldsymbol{h}_{t+1}$ and $\boldsymbol{y}_{t+1}$ by $\boldsymbol{h}_t$, $\boldsymbol{S}_t$ and $\boldsymbol{z}$;
    Sample $\hat{\boldsymbol{S}}_{t+1}$ using $\boldsymbol{y}_{t+1}$;
    $t = t + 1$;
    if $t > t_p$ then
        $\boldsymbol{S}_t = \hat{\boldsymbol{S}}_t$;
    end
end
$t_{EoL} = t$;
Output: return $\boldsymbol{S} = \hat{\boldsymbol{S}}_t,\ t = t_p, \cdots, t_{EoL}$
```

D. Prognostics

Degradation Prediction: After sufficient offline training, Algorithm 1 shows the pseudocode used to make a present-to-EoL prediction at time step t_p, when given previous data collected in $0 < t < t_p$. At time t, the generator takes the transformed feature data $\boldsymbol{S}_t$ as inputs and outputs $\boldsymbol{y}_t$ as the parameters of a probability distribution of the data point $\boldsymbol{S}_{t+1}$. During the prediction process, $\hat{\boldsymbol{S}}_t$ is sampled based on the GMM parameters and categorical distributions at time step t. Unlike during the training process, the predicted $\hat{\boldsymbol{S}}_t$ is fed into the next time step $t+1$. The prediction process will continue until $\mathcal{HI} = (0, 1)$ is achieved.

RUL Prediction: The procedure of RUL prediction is based on the present-to-EoL prediction. Starting from prediction time t_p, the algorithm calculates the prediction until the health state indicator $\mathcal{HI} = (0, 1)$ is obtained at time step t_{EoL}. Then, the predicted RUL is defined as

$$\text{RUL} = t_{\text{EoL}} - t_p \tag{23}$$

where $t_p = 1, 2, \ldots, t_{\text{EoL}}$ represents the RUL prediction and can be carried out at each time step.

IV. Experiments

A. Experimental Setup

Dataset Descriptions: In this article, three types of data (aeroengine, wind turbine, and lithium-ion battery) have been employed to verify the effectiveness and flexibility of the proposed method in different industrial applications.

1) *MAPSS:* The aeroengine data, provided by Modular Aero-Propulsion System Simulation (MAPSS), consist of multiple multivariate run-to-failure recordings (21 sensors

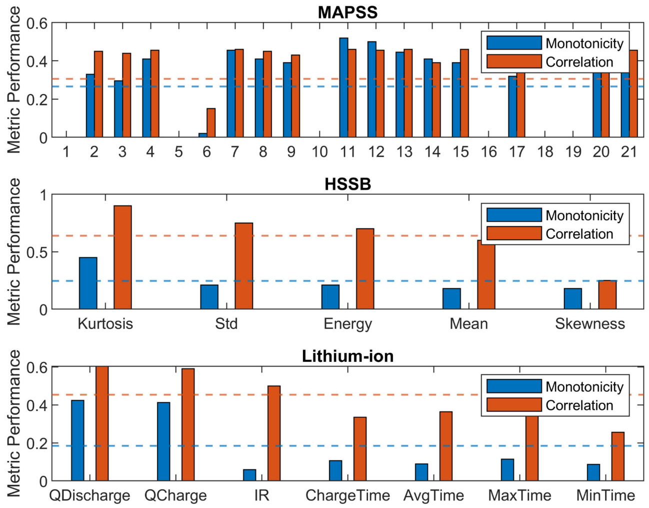

Fig. 3. Correlative and monotonic metrics performance of MAPSS, HSSB, and Lithium-ion. The dot-line is the average value.

and 3 operational settings, see [31]) from a fleet of aero-engines with dissimilar levels of initial wear and unspecified manufacturing disparity. Sensors (e.g., 1st, 5th, 10th, etc.) with constant records are eliminated for contributing nonsensical information. To select the optimal representative degradation feature (i.e., sensor), all features are assessed by correlative and monotonic metrics. As shown in Fig. 3, the 11th (static pressure at HPC outlet), the 12th (ratio of fuel flow to Ps30), and the 13th (corrected fan speed) features produce similar high criteria values. To simplify, the 11th feature is selected.

2) *HSSB:* The wind turbine generator (WTG) vibration data, provided by Green Power Monitoring System, were collected through a 50-days operation of real-world 32222-J2-SKF tapered high-speed shaft bearings (HSSB) installed in a 2.2-MW WTG with a typical shaft speed of 30 Hz, ending with an inner race fault. By assessing the typical time-series statistical features (mean, std, skewness, kurtosis, and energy), kurtosis was found to be the most representative degradation feature, shown in Fig. 3. Visually, it undergoes a growing tendency of decay at an early stage and accelerated growth at the end.

3) *Lithium-ion:* The dataset [32] consists of 124 commercial lithium-ion batteries cycled to failure under various fast-charging conditions. These lithium-ion phosphate (LFP)/graphite cells (1.1 Ah, 3.3 V), manufactured by A123 Systems, were cycled in horizontal cylindrical fixtures on a 48-channel Arbin LBT potentiostat in a 30 °C forced convection temperature chamber. To capture the electrochemical evolution of individual cells during cycling, the cycle-to-cycle evolution of the discharge voltage curve is considered as the most representative degradation feature according to Fig. 3.

In each case, every selected feature is sampled to a fixed sequence length T_s at intervals of $n = \text{ceil}\,(T/T_{\max})$ (where $\text{ceil}(\cdot)$ returns a ceiling value), to form a training dataset. At each time step $t < T_s$, the health indicator $\mathcal{HI} = (1, 0)$ was concatenated to $f_t \in \mathcal{R}^k$ $(k = 1)$ while $\mathcal{HI} = (0, 1)$ at $T_s \leq t \leq T_{\max}$. To facilitate computational efficiency, $T_{\max}$ is regarded as a hyperparameter here and manually set to 100.

Model Layout Details: The LSTM network in generator G consists of 256 internal (hidden) units. The number of components for the GMM is set to $M = 10$. Discriminator D has a bidirectional layout, whereas G is unidirectional.

Baseline Model: Two distinct baseline models have been employed to make a comparative study. By removing the encoder, a pure decoder LSTM is used as a baseline autoregressive model without latent variables, self-trained entirely with loss function $\mathcal{L} = \mathcal{L}_f^G + \mathcal{L}_h^G$ to predict the next status at each time step in the recurrence. The second one is VAE (G) (without adversarial training) with loss function $\mathcal{L} = \mathcal{L}_f^G + \mathcal{L}_h^G + \mathcal{L}_k^G$.

Implementation: Backpropagation through time (BPTT) and mini-batch stochastic gradient descent (SGD) were used [23], with the batch size set to 10. The model was pretrained for 10 epochs with loss function $\mathcal{L}^G = \mathcal{L}_f^G + \mathcal{L}_h^G + \mathcal{L}_k^G$ to balance G and D at the early stage of the training [33], [34]. Layer normalization and recurrent dropout with a keep probability of 90% were applied. The learning rate was set to 0.001 and gradient clipping of 1.0 was used. The fivefold cross-validation is employed for parameter tuning. The implementation was built based on the Tensorflow platform equipped with NVIDIA Geforce GTX 1080 Ti and Titan Xp GPU with 32-GB memory.

Evaluation: The models are compared by performance feedback from prognostics metrics: Prognostic horizon (PH), $\alpha - \lambda$ accuracy, relative accuracy (RA), and convergence. PH is defined as the difference between the EoL and the first time when the prediction result continuously resides in the accuracy zone, which has a constant bound with a magnitude of α error with respect to true EoL. The $\alpha - \lambda$ accuracy determines whether a prediction falls within specified limits (α of the actual RUL) at specific circle t_λ, which is expressed with a fraction of λ between starting cycle of RUL prediction $t_p(\lambda = 0)$ and EoL $(\lambda = 1)$ as

$$t_\lambda = t_p + \lambda(\text{EoL}_{\text{true}} - t_p). \tag{24}$$

RA is the relative accuracy between the true and predicted RUL over α error zone at t_λ, shown in

$$\text{RA} = 1 - \frac{|\text{RUL}_{\text{true}} - \text{RUL}|}{\alpha \text{EoL}_{\text{true}}}, \text{ at } t_\lambda. \tag{25}$$

Convergence [quantified by the center of mass (CoM), (26)] is defined as the Euclidean distance between $(t_p, 0)$ and the centroid (t_C, E_C) of the area under the relative error rate curve $E(k)$ between t_p and EoL

$$\text{CoM} = \sqrt{(t_C - t_p)^2 + E_C^2} \tag{26}$$

with

$$t_C = \frac{1}{2} \frac{\sum_{k=p}^{\text{EoL}} (t_{k+1}^2 - t_k^2) E(k)}{\sum_{k=p}^{\text{EoL}} (t_{k+1} - t_k) E(k)} \tag{27}$$

$$E_C = \frac{1}{2} \frac{\sum_{k=p}^{\text{EoL}} (t_{k+1} - t_k) E(k)^2}{\sum_{k=p}^{\text{EoL}} (t_{k+1} - t_k) E(k)}. \tag{28}$$

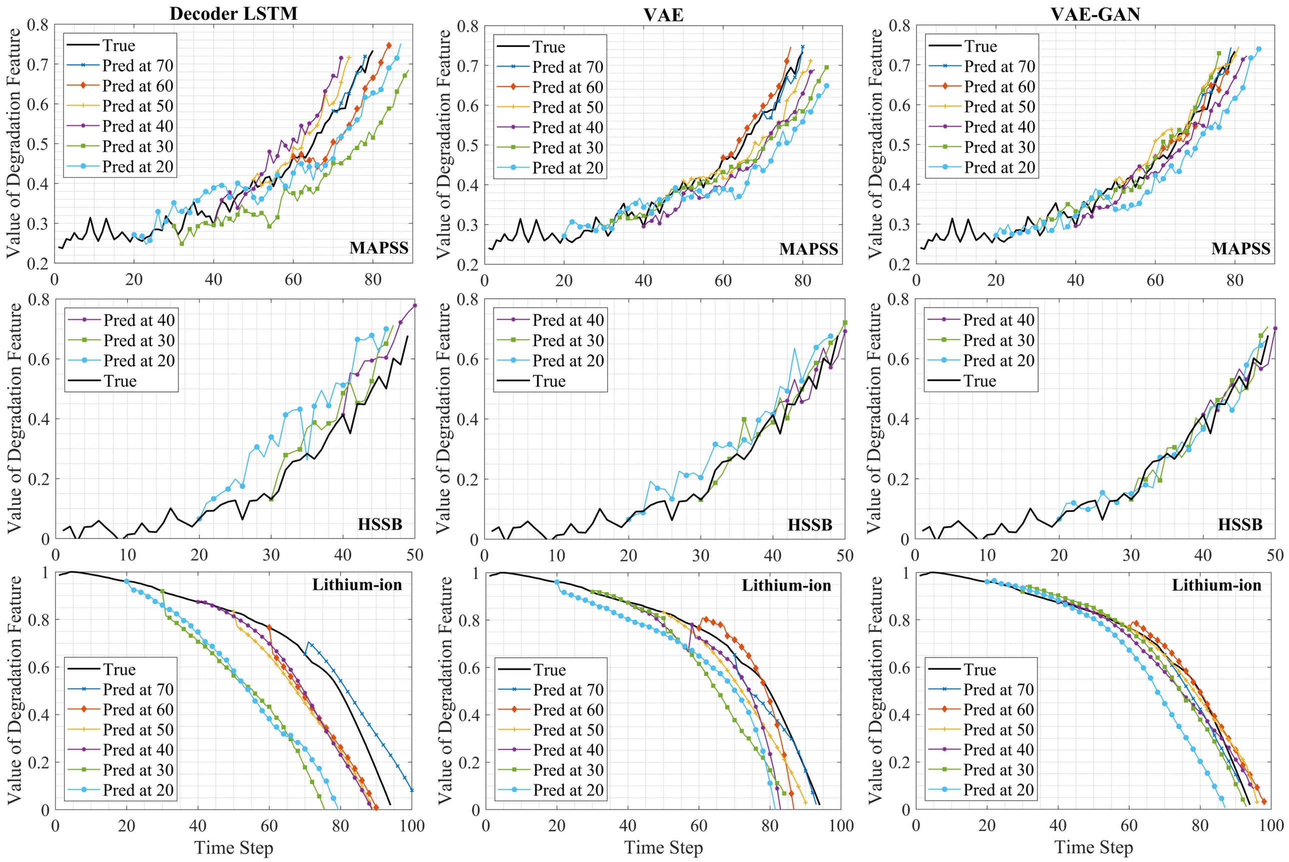

Fig. 4. Present-to-EoL prediction of degradation progress by decoder LSTM, VAE, and VAE-GAN.

The performance of degradation prediction is evaluated using the mean absolute error (MAE) on generated output as

$$\mathrm{MAE}_{\mathrm{decay}} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{T_s}\sum_{t=t_p}^{T_s}\left|\hat{\boldsymbol{S}}_t - \boldsymbol{S}_t\right|\right) \tag{29}$$

where T_s is the length of series, $\hat{\boldsymbol{S}}_t$ is the predicted value, $\boldsymbol{S}_t$ is the ground truth, and N is the number of series.

B. Results and Discussion

The present-to-EoL prediction results for MAPSS, HSSB, and Lithium-ion generated by the baselines (decoder LSTM, VAE) and the proposed VAE-GAN model are shown in Fig. 4 by columns, respectively. In each row, one typical test series for each case is visualized at every 10-time steps as an example. The $\mathrm{MAE}_{\mathrm{decay}}$ for all test series are listed in Table I. The starting circle of prediction t_p is 20 since the indicator remains stable at the beginning. By utilizing decoder LSTM as a standalone predictive model, the degradation prediction can be conditioned on the previous points. Specifically, the decoder LSTM is employed at first to *encode* the observations into a hidden state h. Afterward, h is used as the initial hidden state to yield the remaining degradation prediction. The degradation curves predicted by decoder LSTM roughly represent the real trend, and the degradation distribution becomes closer to the ground truth as the prediction step approaching to EoL. As more observations become available, the result can be more accurate. According to Table I, VAE is able to produce predictions more accurate than decoder LSTM as the $\mathrm{MAE}_{\mathrm{decay}}$ converges faster to a certain level (e.g., $\mathrm{MAE}_{\mathrm{decay}} < 0.02$ in MAPSS) after step 50 ($t > 50$). Furthermore, the smaller $\mathrm{MAE}_{\mathrm{decay}}$ by VAE in the first half of the lifetime indicates that VAE can generate better degradation predictions than decoder LSTM even at the early time. The main reason is that the prediction is conditioned not only on the previous observations but also on the latent vector z encoded by bidirectional LSTM. As Table I shows, VAE-GAN outperforms the baseline models at the early stage as $\mathrm{MAE}_{\mathrm{decay}}$ within 0.05 (e.g., MAPSS), which indicates that adversarial training helps VAE capture the distribution of real data better.

Fig. 5 depicts the RUL predictions of three typical series with the related RUL ground truth as reference, i.e., series #1 (Lithium-ion), series #2 (MAPSS), and series #3 (HSSB). The RUL prediction is visualized at every five time steps. As the PH performance is illustrated in Fig. 5, a longer PH implies more time to take corrective action based on a prediction with some credibility. The allowable error bound α with respect to EoL ground truth is set to 0.05. Evidently, the PH of VAE is wider than that of decoder LSTM, and the VAE-GAN further extends the PH. Both VAE and VAE-GAN provide sufficient PH allowing for corrective maintenance.

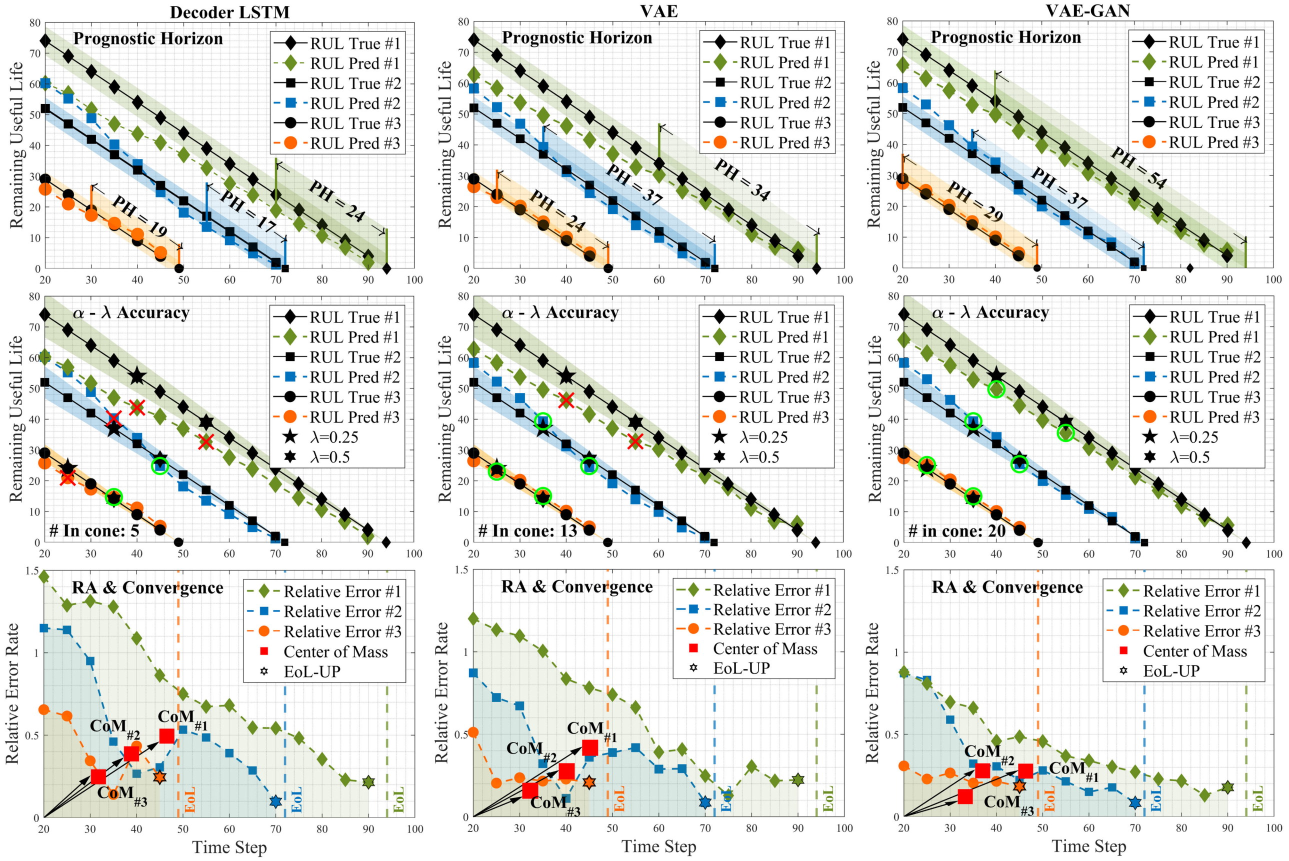

Fig. 5. *Row 1*: PH with $\alpha = 0.05$. It illustrates whether the algorithm predicts within the desired accuracy around EoL and sufficiently in advance. *Row 2*: $\alpha - \lambda$ accuracy performance with $\alpha = 0.1, \lambda = 0.25, 0.5$. This further illustrates if the algorithm stays within desired performance levels relative to RUL at a given time. *Row 3*: Relative error rate (1-RA) and convergence performance with $\alpha = 0.1$. The relative error rate quantifies how well an algorithm performs at a given time relative to RUL. CoM quantifies how fast the performance converges.

The RUL prediction quality evaluated by the $\alpha - \lambda$ metric is illustrated in Fig. 5, which label either "True" or "False" by verifying whether the prediction falls within α ($\alpha = 0.1$) accuracy when prognosticated at early (i.e., $\lambda = 0.25$) or halfway (i.e., $\lambda = 0.5$) to EoL from when the first prediction is made. This is a more stringent requirement than staying within a converging cone of the error margins as a system nears EoL. Since t_λ may not be consistent with the frequency of the prediction step, t'_λ that is closest to t_λ is chosen. It can be observed that VAE and VAE-GAN predict more precise RUL than Decoder LSTM in the early and medium term.

As highlighted by the RULs provided in Fig. 5 and Table I, the predictions by all three models converge to the true RULs, which validate the assumption that prognostic performance improves as more information becomes available with time. Then, the RA metric in (25) is employed to quantify the accuracy levels. The RA by decoder LSTM is relatively higher and fluctuates more heavily than that of VAE. The VAE-GAN further lowers the relative error and flattens the fluctuation, proving to be a more accurate and stable predictive model. Since RA outputs error information at a specific time step, to assess the general error of models, cumulative relative accuracy (CRA), the average of RA values accumulated at every cycle [2], is used to produce an aggregate accuracy level. The average RA at t_λ and CRA of all test series are presented in Table II.

Row 3 in Fig. 5 presents the performance of convergence metric indicating the rate at which the relative accuracy improves with time. As stated earlier, convergence is the Euclidean distance between $(t_p, 0)$ and the centroid of the area under the RA curve from t_p to the End-of-Useful-Predictions (EoUP). EoUP is introduced to express the minimum acceptable PH in demand to take maintenance. From the industrial perspective, any prediction made beyond EoUP is of little or no use since it does not leave enough time to carry out corrective measures. Considering the concept that lower distance implies a faster convergence, it can be seen in Fig. 5 and Table II that, compared to decoder LSTM, VAE and VAE-GAN can produce reliable predictions at earlier stages. Moreover, the convergence of VAE after adversarial training has been slightly improved.

C. Extended Discussion

1) Multivariate Time Series: Theoretically and practically, the proposed model can be easily extended to handle multifeature input. Since the outputs at each time step are the parameters of the probability distribution of the next data $\mathcal{S}_t$, a multivariate normal distribution should be employed in such case. For

TABLE I
MAE PERFORMANCE

MAPSS

Time	Decoder LSTM		VAE		VAE-GAN	
Steps	Decay	RUL	Decay	RUL	Decay	RUL
20	0.084	7.13	0.058	5.82	0.048	4.23
25	0.123	5.96	0.061	6.03	0.046	4.14
30	0.086	7.07	0.040	5.87	0.031	3.45
35	0.087	4.57	0.043	4.46	0.023	3.32
40	0.056	4.56	0.051	4.02	0.032	2.63
45	0.080	3.94	0.046	3.54	0.028	2.56
50	0.043	4.20	0.039	3.78	0.026	2.56
55	0.038	3.86	0.032	3.46	0.016	1.82
60	0.040	3.72	0.021	2.38	0.018	1.54
65	0.035	3.82	0.019	1.90	0.019	1.78
70	0.024	2.91	0.016	1.96	0.017	1.49

HSSB

Time	Decoder LSTM		VAE		VAE-GAN	
Steps	Decay	RUL	Decay	RUL	Decay	RUL
20	0.128	3.21	0.062	2.51	0.033	2.02
25	0.086	3.02	0.054	1.18	0.040	1.12
30	0.070	2.12	0.033	1.36	0.037	1.30
35	0.068	1.34	0.035	1.16	0.036	1.30
40	0.071	1.87	0.049	1.12	0.036	1.05
45	0.052	1.20	0.043	1.04	0.035	1.00

Lithium-ion

Time	Decoder LSTM		VAE		VAE-GAN	
Steps	Decay	RUL	Decay	RUL	Decay	RUL
20	0.249	13.73	0.121	11.28	0.114	8.25
25	0.269	12.10	0.118	10.65	0.102	7.60
30	0.301	12.34	0.144	10.30	0.044	6.54
35	0.292	12.02	0.127	9.44	0.046	6.21
40	0.122	10.22	0.109	8.86	0.046	4.31
45	0.151	8.11	0.098	8.35	0.043	4.56
50	0.173	7.05	0.104	7.16	0.028	4.30
55	0.164	6.33	0.091	6.23	0.027	4.26
60	0.179	6.40	0.083	4.68	0.037	3.50
65	0.097	5.13	0.082	3.84	0.026	3.05
70	0.070	5.10	0.051	3.33	0.028	2.86

TABLE II
(CUMULATIVE) RELATIVE ACCURACY AND CONVERGENCE

MAPSS

t	Decoder LSTM			VAE			VAE-GAN		
	RA	CRA	CoM	RA	CRA	CoM	RA	CRA	CoM
$t_{0.25}$	0.48	0.46	24.81	0.53	0.60	21.72	0.62	0.62	19.18
$t_{0.5}$	0.57			0.65			0.74		

HSSB

t	Decoder LSTM			VAE			VAE-GAN		
	RA	CRA	CoM	RA	CRA	CoM	RA	CRA	CoM
$t_{0.25}$	0.38	0.59	12.94	0.74	0.73	12.62	0.78	0.77	12.28
$t_{0.5}$	0.66			0.78			0.80		

Lithium-ion

t	Decoder LSTM			VAE			VAE-GAN		
	RA	CRA	CoM	RA	CRA	CoM	RA	CRA	CoM
$t_{0.25}$	-0.08	0.22	26.54	0.17	0.38	25.19	0.55	0.57	24.33
$t_{0.5}$	0.33			0.34			0.64		

Note: $\alpha = 0.1; \lambda = 0.25, 0.5.$

example, in order to satisfy two feature series input, a modified probability distribution [compared to (11)] that considering the correlation between two features (instead of i.i.d.) is adopted

$$p(f_t^1, f_t^2) = \sum_{i=1}^{M} \pi_i \mathcal{N}(f_t^1, f_t^2 | \mu_i^1, \sigma_i^1, \mu_i^2, \sigma_i^2, \rho_i^{12}) \quad (30)$$

TABLE III
MULTIVARIATE FEATURE SERIES INPUT PERFORMANCE COMPARISON

MAPSS by VAE-GAN

t_λ	$\boldsymbol{f}_t \subseteq \mathcal{R}^1$			$\boldsymbol{f}_t \subseteq \mathcal{R}^2$			$\boldsymbol{f}_t \subseteq \mathcal{R}^3$		
	RA	CRA	CoM	RA	CRA	CoM	RA	CRA	CoM
$t_{0.25}$	0.62	0.62	19.18	0.68	0.64	18.02	0.13	-0.05	22.78
$t_{0.5}$	0.74			0.77			-0.02		

Note: $\alpha = 0.1; \lambda = 0.25, 0.5.$

where $\mathcal{N}$ is the probability distribution function for a bivariate normal distribution, and ρ_i^{12} is the correlation parameter of each bivariate normal distribution.

Accordingly, the output $\boldsymbol{y}_t$ (10) is modified as

$$\begin{aligned} \boldsymbol{y}_t = \boldsymbol{W}_y \boldsymbol{h}_t + \boldsymbol{b}_y = [&(\hat{\pi}_1, \mu_1^1, \hat{\sigma}_1^1, \mu_1^2, \hat{\sigma}_1^2, \hat{\rho}_1^{12}), \ldots, \\ &(\hat{\pi}_M, \mu_M^1, \hat{\sigma}_M^1, \mu_M^2, \hat{\sigma}_M^2, \hat{\rho}_M^{12}), \\ &(\hat{p}_1, \hat{p}_2)] \end{aligned} \quad (31)$$

where $\boldsymbol{y_t} \subseteq \mathcal{R}^{6M+2}$ can be split into M mixed Gaussian distributions to describe $f_t = (f_t^1, f_t^2)$ and one categorical (p_1, p_2) distribution to describe health indicator $\mathcal{HI}$. In addition to exp and softmax operations (12)–(14), tanh operation is applied to ρ to ensure $\rho \subseteq [-1, 1]$.

Identically, a trivariate normal distribution (with $\boldsymbol{y_t} \subseteq \mathcal{R}^{10M+2}$) should be considered for three feature series input ($\boldsymbol{f}_t \subseteq \mathcal{R}^3$), and an n-variate normal distribution (with $\boldsymbol{y_t} \subseteq \mathcal{R}^{(1+2n+C_n^2)M+2}$) for n-dimensional feature ($\boldsymbol{f}_t \subseteq \mathcal{R}^n$). The general formula for the n-dimensional normal density is

$$\mathcal{F}_{\underline{f}}(f^1, f^2, \ldots, f^n) = \frac{\exp\{-\frac{1}{2}(\underline{f} - \underline{\mu})' K^{-1} (\underline{f} - \underline{\mu})\}}{(\sqrt{2\pi})^n \sqrt{\det(K)}} \quad (32)$$

where $\underline{f} = (f^1, \ldots, f^n)$, $\underline{\mu} = E(\underline{f})$ and K is the covariance matrix.

To further explore the relationship between prognostic accuracy and feature dimension, the proposed model is extended to match feature series with 2 or 3 dimensions, where the top n most representative features are selected based on Section III-A. Features 1, 2, and 3 represent the 11th (static pressure at HPC outlet), 12th (ratio of fuel flow to Ps30), and 13th (corrected fan speed) features, respectively. The comparative result is presented in Fig. 6 and Table III. It can be observed that there is a slight increase in performance when $\boldsymbol{f}_t \subseteq \mathcal{R}^2$. However, when the feature dimension reaches 3, it dramatically deteriorates the prognostic performance. This is because the added feature 2 is highly correlated with feature 1 and offers little useful information (also known as feature redundancy) in terms of training. Additionally, using multiple inputs significantly increases the computational burden and, thus, is hard to achieve Nash equilibrium during the training process. When the feature dimension exceeds 3, the training process becomes unstable and cannot guarantee convergence. One of the reasons is that each feature dimension has a different distribution, even under the same degradation mode. It is hard to model all the different distributions at the same time precisely. In the future work, a multivariate adaptive prognostic model by generative adversarial learning will be further studied.

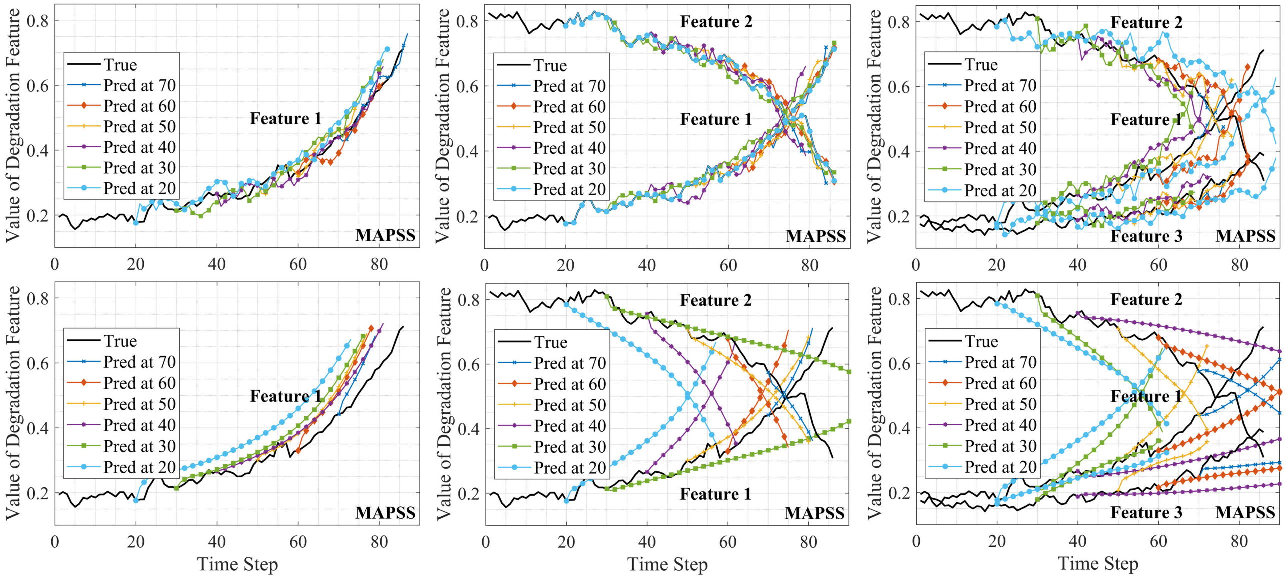

Fig. 6. *Row 1*: The present-to-EoL prediction of degradation by VAE-GAN of multifeatures. *Row 2*: Performance by VAE-GAN without GMM part.

TABLE IV
PERFORMANCE OF THE PROPOSED METHOD WITH/WITHOUT GMM

MAPSS

Model	GMM	-	Prediction time steps					
			20	30	40	50	60	70
VAE-GAN	×	Decay	0.068	0.053	0.052	0.044	0.041	0.033
		RUL	7.72	6.09	4.36	4.18	4.20	3.72
	✓	Decay	0.048	0.031	0.032	0.026	0.018	0.017
		RUL	4.23	3.45	2.63	2.56	1.54	1.49

2) Ablation Study: To evaluate the effects of GMM, the GMM part is removed in the proposed VAE-GAN. In order to meet the one-hot encoded constraints of $\mathcal{HI}$, the exp and softmax operations are attached to the fully connection layer. The performance on the MAPSS dataset is shown in Table IV and illustrated in column 1 of Fig. 6. Here, the model is learning different degradation modes, where different conditions may or may not occur, thus averaging over different events is not meaningful. In this case, by integrating the GMM, the performance improvements indicate that multiple subdistributions help model varied degradation mechanisms. In other words, degradations have multiple possible modes that should not be mixed or averaged. The effect of GMM on multivariate feature modeling is also conducted as illustrated in row 2 of Fig. 6, and the poor performance without GMM further validates that Gaussian components have two complementary roles as proposed and proofed in [35]: 1) separately modeling different stochastic events, and 2) separately modeling scenarios governed by different rules.

3) Threshold: In most prognostic methods, EoL is obtained when an indicator (e.g., selected feature) exceeds a predefined threshold. However, this will introduce additional concerns. Take CMAPSS data as an example, if the averaged feature value (11th feature) at failure time is chosen as a threshold among 100 degradation cases, there is a 50% probability that the system will break down before reaching that threshold. If the minimum feature value at failure time is chosen, this will introduce a systematic error (± 13.61) on RUL prediction even if the degradation prediction is 100% accurate. In this article, the feature transformation method, which incorporates one-hot health indicator, enables the model to learn different EoLs and, thus, bypass low-accuracy prediction produced from an imprecise predefined failure threshold.

4) Generalization: Like most of the classical machine learning-based prognostic models, the proposed method in this article needs enough run-to-failure historical data to achieve a significant performance level. Although the implementation of Gaussian components in our proposed model enables separately modeling different stochastic events and separately modeling scenarios governed by different rules [35], it is unable to produce an accurate prediction for new coming data that have large variations from the learning datasets (i.e., data follow a new degradation mode that has not been observed before). To tackle this problem, concepts such as physics-informed [36] or domain adaptation [37] are encouraged.

V. CONCLUSION

This article proposed a novel sequence-to-sequence VAE with an adversarial learning approach to predict long-term degradation progress and RUL without defining a specific failure threshold. This approach used correlative and monotonic metrics to identify the critical features related to degradation, which are then concatenated with health indicator vectors to train the model. The VAE consisted of a bidirectional LSTM-based encoder and an autoregressive LSTM-GMM-based decoder, fully utilizing the capability of LSTM in learning long-term dependencies in time series data. The output of the LSTM within the decoder was connected to a fully connected layer to map the output into the parameters of a GMM and a categorical distribution for sampling consequent predictions. Experiments on real-world health monitoring data of aeroengines, wind turbines, and lithium-ion batteries verified the effectiveness and robustness of the proposed approach. Prediction is conditioned on the previous encoded observations, which enables multimode degradation prediction. The adversarial training helps the VAE better capture the distribution of real degradation progress, thus leading to a more accurate RUL prediction.

References

[1] Y. Jiang and S. Yin, "Recent advances in key-performance-indicator oriented prognosis and diagnosis with a MATLAB toolbox: DB-KIT," *IEEE Trans. Ind. Informat.*, vol. 15, no. 5, pp. 2849–2858, May. 2018.

[2] N.-H. Kim, D. An, and J.-H. Choi, *Prognostics and Health Management of Engineering Systems*. Cham, Switzerland: Springer, 2017.

[3] D. Goodman, J. P. Hofmeister, and F. Szidarovszky, *Prognostics and Health Management: A Practical Approach to Improving System Reliability Using Condition-Based Data*. Hoboken, NJ, USA: Wiley, 2019.

[4] J. Lee, H.-A. Kao, and S. Yang, "Service innovation and smart analytics for Industry 4.0 and big data environment," *Procedia CIRP*, vol. 16, pp. 3–8, 2014.

[5] S. Yin, J. J. Rodriguez-Andina, and Y. Jiang, "Real-time monitoring and control of industrial cyberphysical systems: With integrated plant-wide monitoring and control framework," *IEEE Ind. Electron. Mag.*, vol. 13, no. 4, pp. 38–47, Dec. 2019.

[6] Y. Jiang, S. Yin, and O. Kaynak, "Data-driven monitoring and safety control of industrial cyber-physical systems: Basics and beyond," *IEEE Access*, vol. 6, pp. 47 374–47384, 2018.

[7] M. Elforjani and S. Shanbr, "Prognosis of bearing acoustic emission signals using supervised machine learning," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5864–5871, Jul. 2018.

[8] X. Li, Q. Ding, and J.-Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Rel. Eng. Syst. Saf.*, vol. 172, pp. 1–11, 2018.

[9] B. Y. Bejarbaneh, E. Y. Bejarbaneh, A. Fahimifar, D. J. Armaghani, M. Z. A. Majid, and M. F. M. Amin, "Intelligent modelling of sandstone deformation behaviour using fuzzy logic and neural network systems," *Bull. Eng. Geol. Environ.*, vol. 77, no. 1, pp. 345–361, 2018.

[10] Y. Wu, M. Yuan, S. Dong, L. Lin, and Y. Liu, "Remaining useful life estimation of engineered systems using vanilla LSTM neural networks," *Neurocomputing*, vol. 275, pp. 167–179, 2018.

[11] F. O. Heimes, "Recurrent neural networks for remaining useful life estimation," in *Proc. IEEE Prognostics Health Manage. Int. Conf.*, 2008, pp. 1–6.

[12] D. An, J.-H. Choi, and N. H. Kim, "Remaining useful life prediction of rolling element bearings using degradation feature based on amplitude decrease at specific frequencies," *Struct. Health Monit.*, vol. 17, no. 5, pp. 1095–1109, 2018.

[13] A. S. S. Vasan, B. Long, and M. Pecht, "Diagnostics and prognostics method for analog electronic circuits," *IEEE Trans. Ind. Electron.*, vol. 60, no. 11, pp. 5277–5291, Nov. 2013.

[14] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[15] A. Soualhi, H. Razik, G. Clerc, and D. D. Doan, "Prognosis of bearing failures using hidden Markov models and the adaptive neuro-fuzzy inference system," *IEEE Trans. Ind. Electron.*, vol. 61, no. 6, pp. 2864–2874, Jun. 2014.

[16] W. Qiao, P. Zhang, and M.-Y. Chow, "Condition monitoring, diagnosis, prognosis, and health management for wind energy conversion systems," *IEEE Trans. Ind. Electron.*, vol. 62, no. 10, pp. 6533–6535, Oct. 2015.

[17] L. Liao, "Discovering prognostic features using genetic programming in remaining useful life prediction," *IEEE Trans. Ind. Electron.*, vol. 61, no. 5, pp. 2464–2472, May. 2014.

[18] T. Gerber, N. Martin, and C. Mailhes, "Time-frequency tracking of spectral structures estimated by a data-driven method," *IEEE Trans. Ind. Electron.*, vol. 62, no. 10, pp. 6616–6626, Oct. 2015.

[19] R. Zhao, D. Wang, R. Yan, K. Mao, F. Shen, and J. Wang, "Machine health monitoring using local feature-based gated recurrent unit networks," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1539–1548, Feb. 2018.

[20] A. S. Yoon *et al.*, "Semi-supervised learning with deep generative models for asset failure prediction," 2017, *arXiv:1709.00845*.

[21] J. Li, S. Liu, H. He, and L. Li, "A novel framework for gear safety factor prediction," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 1998–2007, Apr. 2019.

[22] O. Mogren, "C-RNN-GAN: Continuous recurrent neural networks with adversarial training," 2016, *arXiv:1611.09904*.

[23] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[24] D. Ha and J. Schmidhuber, "Recurrent world models facilitate policy evolution," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 2450–2462.

[25] F. Camci, K. Medjaher, N. Zerhouni, and P. Nectoux, "Feature evaluation for effective bearing prognostics," *Qual. Rel. Eng. Int.*, vol. 29, no. 4, pp. 477–486, 2013.

[26] P. Sedgwick, "Pearson's correlation coefficient," *BMJ*, vol. 345, 2012, Art. no. e4483

[27] L. Guo, N. Li, F. Jia, Y. Lei, and J. Lin, "A recurrent neural network based health indicator for remaining useful life prediction of bearings," *Neurocomputing*, vol. 240, pp. 98–109, 2017.

[28] F. Yang, M. S. Habibullah, T. Zhang, Z. Xu, P. Lim, and S. Nadarajan, "Health index-based prognostics for remaining useful life predictions in electrical machines," *IEEE Trans. Ind. Electron.*, vol. 63, no. 4, pp. 2633–2644, Apr. 2016.

[29] D. Ha and D. Eck, "A neural representation of sketch drawings," 2017, *arXiv:1704.03477*.

[30] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.

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

[32] K. A. Severson *et al.*, "Data-driven prediction of battery cycle life before capacity degradation," *Nat. Energy*, vol. 4, no. 5, 2019, Art. no. 383.

[33] H. Ham, T. J. Jun, and D. Kim, "Unbalanced GANs: Pre-training the generator of generative adversarial network using variational autoencoder," 2020, *arXiv:2002.02112*.

[34] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.

[35] K. O. Ellefsen, C. P. Martin, and J. Torresen, "How do mixture density RNNs predict the future?" 2019, *arXiv:1901.07859*.

[36] Y. A. Yucesan and F. A. Viana, "A physics-informed neural network for wind turbine main bearing fatigue," *Int. J. Prognostics Health Manage.*, vol. 11, no. 1, 2020, Art. no. 17.

[37] X. Li, W. Zhang, N.-X. Xu, and Q. Ding, "Deep learning-based machinery fault diagnostics with domain adaptation across sensors at different places," *IEEE Trans. Ind. Electron.*, vol. 67, no. 8, pp. 6785–6794, Aug. 2020.

Yu Huang (Student Member, IEEE) received the B.S. and M.S. degrees in aeronautics and astronautics engineering from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2015 and 2018, respectively. He is currently working toward the Ph.D. degree in electrical engineering with Florida Atlantic University, Boca Raton, FL, USA.

His research interests include prognostics and health management, machine learning, and its applications in smart grid and oceanography.

Yufei Tang (Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Rhode Island, Kingston, RI, USA, in 2016.

He is currently an Assistant Professor with the Department of Computer and Electrical Engineering & Computer Science, Florida Atlantic University, Boca Raton, FL, USA. His research interests include machine learning, big data analytics, and sustainability for energy and environment.

Dr. Tang is an Early-Career Research Fellow of the National Academies Gulf Research Program in 2019. He is a also a recipient of several other awards, including the Steve Bouley and Rhonda Wilson Graduate Fellowship Award in 2016, the Chinese Government Award for Outstanding Student Abroad in 2016, the IEEE PESGM Graduate Student Poster Contest, Second Prize, in 2015, and the IEEE International Conference on Communications Best Paper Award in 2014.

James VanZwieten received the B.S., M.S., and Ph.D. degrees in ocean engineering from Florida Atlantic University, Boca Raton, FL, USA, in 2001, 2003, and 2007, respectively.

He is currently an Associate Research Professor with the Department of Civil, Environmental, and Geomatics Engineering, Florida Atlantic University. He was an Assistant Research Professor with the Southeast National Marine Renewable Energy Center operated by Florida Atlantic University. His research interests include modeling and control of marine vehicles, in-stream hydrokinetic energy production, ocean thermal energy conversion, and seawater air conditioning.

Dr. VanZwieten is a member of the American Society of Civil Engineers Marine Renewable Energy Committee and the Chair of its In-Stream Hydrokinetic Subcommittee.