An Adversarial Learning Approach for Machine Prognostic Health Management

Yu Huang[1], Yufei Tang[1]*, James VanZwieten[1], Jianxun Liu[2] and Xiaocong Xiao[2]
[1]*Florida Atlantic University Boca Raton, FL 33431, USA*
[2]*Hunan University of Science and Technology, Hunan, Xiangtan 411201, China*
{yhwang2018, tangy, jvanzwi}@fau.edu, {liujx, xiaoxc}@hnust.edu.cn

***Abstract*—Achieving accurate remaining useful life (RUL) prediction for prognostic and health management (PHM) depends upon sufficient prior degradation apprehension of critical components within the system. However, such prior knowledge is not always readily available in practice. We alleviate this shortcoming by proposing a novel data-driven framework that is capable of providing accurate RUL prediction without the need for any prior failure threshold knowledge. Correlative and monotonic metrics are utilized to identify critical features throughout the degradation progress. Subsequently, we append one-hot health state indicators to extracted degrading features, which are utilized together as adversarial training data for a Long Short-Term Memory (LSTM) network-based model. Finally, we utilize a fully-connected layer to project the LSTM outputs into the parameters of a Gaussian mixture model (GMM) in conjunction with a categorical distribution, from which the long-term degradation progress is sampled. We verify the performance of the proposed framework using aeroengine health data simulated by Modular Aero-Propulsion System Simulation (MAPSS), and the results demonstrate that significant performance improvement can be achieved for long-term degradation progress and RUL prediction tasks.**

Keywords*—Generative Adversarial Learning, Long Short-Term Memory, Prognostics and Health Management, Remaining Useful Life Prediction, Gaussian Mixture Model

I. Introduction

Machine condition monitoring and intelligent maintenance is a vital part in many industries including energy, automotive, aerospace and heavy industries [1]. Conventional schemes, akin to corrective maintenance and scheduled preventive maintenance, are inadequate for meeting the increasing industrial demand for efficiency and reliability. Recent developments in prognostics and health management (PHM) enable real-time system health condition assessment along with the prediction of its subsequent status utilizing up-to-the-minute information, making it a promising technology [2].

To extract practical information and make suitable maintenance decisions from historical and online statistics, data-driven approaches are regarded as powerful solutions for PHM [3]. Machine learning, in which a model is trained to yield a desired output, such as the status of degradation or life cycle, is capable of harnessing industrial big data for maintenance applications. For example, neural network-based approaches have been proposed that map monitored feature data to machine health conditions [4], [5]. The combination of neural networks and fuzzy systems have been successfully employed by several researchers to apprehend more knowledge for prognostics [6]. To account for time dependency between sequential data, a moving time-window based feature extraction method along with a k-means filter was proposed in [7] to enhance the performance of a multi-layer perceptron model. Recently, a vanilla long short-term memory (LSTM) network and feature extraction method was utilized in [8] for the efficient RUL prediction of intricate industrial system.

In the approaches discussed above, each sampled data is required to associate with a corresponding RUL label as the objective, and the piece-wise linear method [9] is routinely adopted for labeling. If the available label information is relatively small, the advantage of the data-driven approach using machine learning could be highly limited. To overcome this shortcoming, a generative prognostic model with sparse health status information has been built in [10] for the prediction of future asset reliability. Further, a generative adversarial networks based model was established in [11] to cope with the insufficiency of meaningful data for vehicle transmission gear health monitoring. For data-driven methods, remaining useful life (RUL) is attained when a health indicator exceeds a pre-defined failure threshold, which usually needs to be defined in advance. In addition, when support vector machine (SVM) technology is combined, domain knowledge is required to define an appropriate threshold to separate the hyperplane of the high dimensional fault-related features.

Inspired by previous work, this paper proposes a novel data-driven framework based on generative adversarial networks (GAN) [12], aims to enhance the predictions of long-term degradation and remaining useful life without defining any prior failure threshold. Apart from many other applications and developments such as network embedding [13] and image processing [14], this paper is the first work that attempts to use GAN exclusively for machine health prognostics. The main contributions of this research are three-fold: 1) We propose a data-driven framework based on adversarial learning for long-term degradation and remaining useful life prediction. 2) We use a LSTM-based generator network fed with time-series data combined with a one-hot health state indicator to bypass the low accuracy prediction yields produced by imprecise knowledge of predefined failure threshold. 3) We quantify the effectiveness of the proposed framework using aeroengine health data generated by the Modular Aero-Propulsion System

978-1-7281-0467-6/19/$31.00

Simulation (MAPSS) [15].

The remainder of this paper is arranged as follows: Section II gives a brief introduction of the preliminaries of this work. Section III explicitly formulates the problem and presents the adversarial learning framework in detail. Section IV presents the experimental results of the proposed approach as applied to aircraft turbofan engine data. Finally, Section V contains concluding remarks and future research directions.

II. Preliminaries

A. Long Short-Term Memory Networks

PHM research predominantly engage in degradation progress modeling and remaining useful life prediction. In these prognostics cases, available information is usually presented in the form of sequential data, for example time-series sensor records. Therefore, prognosis is commonly formulated as a sequence learning problem that can be solved by constructing related predictive models.

The long short-term memory (LSTM) neural network is a symbolic branch of recurrent neural networks (RNN) which is often used to model sequences of data. The fundamental concept of the LSTM architecture lies in a memory block used to sustain its inherent state over time, and non-linear gate modules that govern dataflow through the block. The LSTM block gives prominence to three gates (i.e., input, output and forget), the constant error carousel (CEC), the $tanh$ activation function, and peephole connections (from cell to gates) [16], [17]. The output of the block is recurrently transferred backward to block input and the three gates, which makes LSTM capable of learning long-term dependencies and therefore is regard as a sophisticated and promising prognostic algorithm.

B. Generative Adversarial Networks

Generative adversarial networks (GANs) [12], devised with the aspiration of originating realistic data, consist of two adversarial models: a generator G that apprehends data distribution, and a discriminator D that appraises the probability of a sample being derived from the training data or G. Both G and D customarily use a non-linear mapping function, for instance, multilayer perceptron (MLP). To grasp a generative distribution p_g that is analogous to training data $\boldsymbol{x}$, the generator formulates a mapping function that projects a prior noise distribution $p_z(\boldsymbol{z})$ to the data space $G(\boldsymbol{z};\boldsymbol{\theta}_g)$. To differentiate the $G(\boldsymbol{z};\boldsymbol{\theta}_g)$ from $D(\boldsymbol{x};\boldsymbol{\theta}_d)$, the discriminator uses an individual scalar depicting the probability that $\boldsymbol{x}$ is derived from the training data p_{data} instead of p_g.

The D is trained to maximize the probability of distinguish real samples from generated samples, while G is simultaneously trained to minimize the log loss $log(1-D(G(\boldsymbol{z})))$. The optimization of GAN can be formulated as a minmax problem with a global optimum for $p_g = p_{data}$ as follows:

$$\min_G \max_D V(D,G) = \begin{aligned} & E_{x\sim p_{data}(\boldsymbol{x})}\left[logD(\boldsymbol{x})\right] + \\ & E_{z\sim p_z(\boldsymbol{z})}\left[log(1-D(G(\boldsymbol{z})))\right] \end{aligned} \tag{1}$$

Inspired by this paradigm, the feasibility of using adversarial training for LSTM networks with continual time-series sensor data will be investigated in this paper.

III. Methodology

A. Problem Statement

In our problem, we have multivariate run-to-failure data from sensor measurements. Let $x^{(i;j)}$ denote the measurement value of sensor j at time index i. Furthermore, let $\boldsymbol{x}^{(i)}$ denote a vector of multivariate sensor measurement such that $\boldsymbol{x}^{(i)} = [x^{(i;1)}, \cdots, x^{(i;m)}]$, where m is the number of sensors. Formally, we describe a sensor measurement matrix by denoting $\mathcal{X} = \{\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(n)}\}$, where $\boldsymbol{x}^{(i)} \subseteq \mathcal{R}^m$ and n is the total time steps of observation. We also denote the corresponding RUL by $\mathcal{Y} = \{\boldsymbol{y}^{(1)}, \cdots, \boldsymbol{y}^{(n)}\}$. In general, the data-driven prognostics approach is to learn the best predictor of run-to-failure degradation from previously observed data, i.e., training data set $\mathcal{D}^T$ sampled from $\mathcal{X}$. Then, based on the predictor model, we can calculate the RUL or equivalent health indicators.

This problem can be formulated as achieving a non-linear mapping function $\mathcal{F}: \boldsymbol{x}' \rightarrow \boldsymbol{z}$, with a latent variable $\boldsymbol{z} \subseteq \mathcal{R}^k$ and $m < k$. Then, the optimal predictor can be defined by a function of $\boldsymbol{z}$ as:

$$f_{\boldsymbol{\gamma}}(\boldsymbol{z}) = \arg\max_{\boldsymbol{y}} p(\boldsymbol{y}\,|\boldsymbol{z},\boldsymbol{\gamma}) \tag{2}$$

where γ is the parameter of the nonlinear mapping function that needs to be optimized through training $f_{\boldsymbol{\gamma}}(\boldsymbol{z})$. The primary goal of this paper is to develop a deep-learning based approach to learn the non-linear mapping function $f_{\boldsymbol{\gamma}}$ for degradation progress modeling and stable RUL prediction given an available training dataset $\mathcal{D}^T$.

B. Proposed Approach

Feature Extraction & Reconfiguration: As shown in Fig. 1, feature extraction and selection is an essential part of prognostics. In this process, we seek to identify critical features that contain sufficient degradation signatures from the original data, in order to increase the efficiency and reliability of prognostics by reducing the cost of feature measurement (i.e., less measurement) and minimizing the dimensions of data (i.e., lower dimension) required to describe the degradation progress. In this work, in consideration of the reality that during the degradation process some critical features either increase or decrease monotonically, correlative and monotonic metrics [18] are first employed to extract the most sensitive features from the entire initially sampled data. Then, instead of using these selected features directly, we utilize a novel data arrangement to represent the machinery degradation progress as follows:

$$\boldsymbol{S}_t = (\boldsymbol{f}_t, \mathcal{HI}) \tag{3}$$

where $\boldsymbol{f}_t \subseteq \mathcal{R}^k$, k is the number of features that have been selected, and $\mathcal{HI} = (h_1, h_2)$ is the health state indicator generated by one-hot encoding from which the model learns the end of life cycle. Specifically, $(1,0)$ signifies the system is

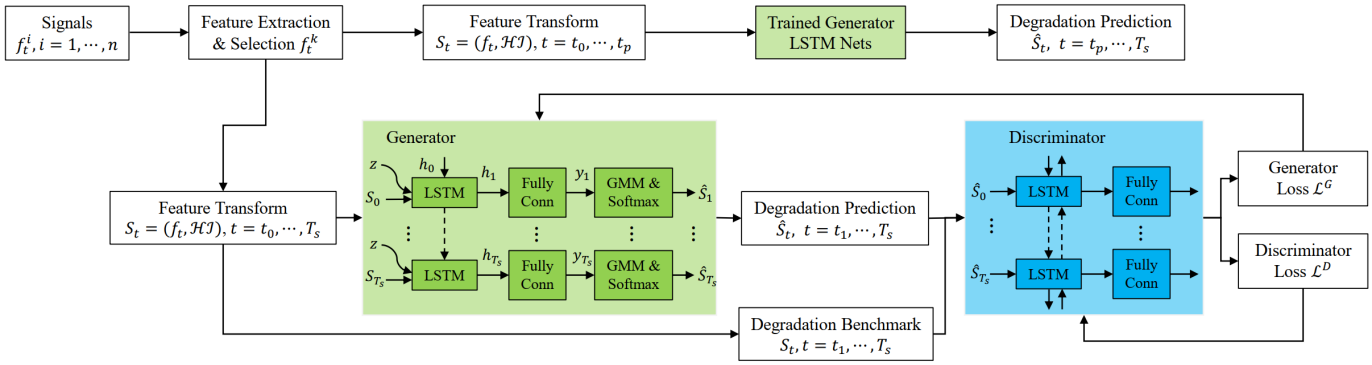


Fig. 1. The proposed generative adversarial learning framework LSTM-GAN for remaining useful life prediction. The Generator G produces sequences of continual degradation prediction. The Discriminator D is trained to differentiate the degradation benchmark from the prediction.

in good condition and is presently functioning, whereas $(0, 1)$ signifies that the system is broken down and stands in need of maintenance. The initial feature is set as $\boldsymbol{S}_0 = (0, 1, 0)$. In addition, the min-max normalization along with mean filtering have been employed in advance to lessen the impact of noise.

LSTM with Adversarial Training: The core of the proposed model is a LSTM neural network with adversarial training (LSTM-GAN). Specifically, the generator G and discriminator D are two distinct recurrent neural networks as shown in Fig. 1. At each time step t, the generator takes in the previous point $\boldsymbol{S}_{t-1}$ and the latent vector $\boldsymbol{z}$ as a concatenated input $\boldsymbol{x}_{t-1}$. The computation of the generator G can be interpreted as follows:

$$\boldsymbol{h}_t = \mathcal{LSTM}\left(\boldsymbol{x}_{t-1}, \boldsymbol{h}_{t-1}\right) \tag{4}$$

where the model input are $\boldsymbol{x}_{t-1}$ and $\boldsymbol{h}_{t-1}$, and the output at each time step is the hidden state $\boldsymbol{h}_t$, which are parameters for a probability distribution of the consequent data $\boldsymbol{S}_t$.

A fully-connected layer is adopted to map the hidden state $\boldsymbol{h}_t$ to the yield vector $\boldsymbol{y}_t$ that could be divided into M mixed Gaussian distributions to describe f_t and one categorical (p_1, p_2) distribution to describe health state indicator $\mathcal{HI}$ as follows:

$$\begin{aligned} \boldsymbol{y}_t &= \boldsymbol{W}_y \boldsymbol{h}_t + \boldsymbol{b}_y \\ &= [(\hat{\pi}_1, \mu_1, \hat{\sigma}_1), \cdots, (\hat{\pi}_M, \mu_M, \hat{\sigma}_M), (\hat{p}_1, \hat{p}_2)] \end{aligned} \tag{5}$$

At each time step, the feature f_t in $\boldsymbol{S}_t$ described by the GMM with M normal distributions is given by:

$$p(f_t) = \sum_{i=1}^{M} \pi_i \mathcal{N}(f_t | \mu_i, \sigma_i) \tag{6}$$

where μ_i is the mean of the ith normal distribution with a standard deviation of σ_i. We should notice that $\sum_{i=1}^{M} \pi_i = 1$.

The exp and $softmax$ operations are adopted since the probability properties and the non-negative of the standard deviations. The probabilities for the categorical distributions are calculated using the outputs as logit values as:

$$\sigma_i = exp(\hat{\sigma}_i) \tag{7}$$

$$\pi_k = \frac{exp(\hat{\pi}_k)}{\sum_{i=1}^{M} exp(\hat{\pi}_i)}, k = 1, 2, \cdots, M \tag{8}$$

$$p_k = \frac{exp(\hat{p_k})}{\sum_{i=1}^{M} exp(\hat{p}_i)}, k = 1, 2 \tag{9}$$

The discriminator consists of a bidirectional LSTM neural network, allowing it to take the time-series in both directions into account for its decisions [19]. The output from each LSTM block in D is fed into a fully-connected layer with sharing weights through time, and the averaged sigmoid output of all the LSTM blocks represents the final decision for the sequence.

The training process is to optimize the loss function which will be given below. First, given generator G, we describe the optimization of the discriminator D. Specifically, the training process of the discriminator involves minimizing the cross entropy, which is similar to the training of sigmoid function-based classifiers. The loss function $\mathcal{L}^D$ is formulated as below:

$$\mathcal{L}^D(\boldsymbol{\theta}_d, \boldsymbol{\theta}_g) = \frac{1}{m} \sum_{i=1}^{m} [logD(\boldsymbol{S}_i) + log(1 - D(G(\boldsymbol{x}_i)))] \tag{10}$$

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

Then, given discriminator D, we optimize G to minimize the discrimination accuracy of D. The reconstructed loss function is formulated as $\mathcal{L}^G = \mathcal{L}_g^G + \mathcal{L}_f^G + \mathcal{L}_h^G$ with:

$$\mathcal{L}_g^G = \frac{1}{m} \sum_{i=1}^{m} [log(1 - D(G(\boldsymbol{x}_i)))] \tag{11}$$

$$\mathcal{L}_f^G = -\frac{1}{m} \sum_{i=1}^{m} log\left(\sum_{k=1}^{M} \pi_{i,k} \mathcal{N}(f | \mu_{i,k}, \sigma_{i,k})\right) \tag{12}$$

$$\mathcal{L}_h^G = \frac{1}{m} \sum_{i=1}^{m} \left[h_1^i log(p_1^i) + h_2^i log(p_2^i)\right] \tag{13}$$

where $\mathcal{L}_g^G$ is the adversarial loss, $\mathcal{L}_f^G$ corresponds to the features and $\mathcal{L}_h^G$ corresponds to the health state indicator.

Algorithm 1: Generative adversarial nets training by Minibatch stochastic gradient descent (SGD).

Initialization: k, $\boldsymbol{\theta}_d$, $\boldsymbol{\theta}_g$ and m
for *the total iteration steps* **do**
  for k *iteration steps* **do**
    Randomly shuffle minibatch of m degradation samples $\boldsymbol{S}^{(1)}, \cdots, \boldsymbol{S}^{(m)}$ from real data distribution $p_{data}(\boldsymbol{S})$;
    Randomly shuffle minibatch of m degradation prediction samples $\hat{\boldsymbol{S}}^{(1)}, \cdots, \hat{\boldsymbol{S}}^{(m)}$ from prior noise distribution $p_g(\hat{\boldsymbol{S}})$;
    Update the parameters of discriminator via ascending stochastic gradient: $\bigtriangledown_{\boldsymbol{\theta}_d} \mathcal{L}^D(\boldsymbol{\theta}_d, \boldsymbol{\theta}_g)$;
  end
  Randomly shuffle minibatch of m degradation prediction samples $\hat{\boldsymbol{S}}^{(1)}, \cdots, \hat{\boldsymbol{S}}^{(m)}$ from prior noise distribution $p_g(\hat{\boldsymbol{S}})$;
  Update the parameters of generator via descending its stochastic gradient: $\bigtriangledown_{\boldsymbol{\theta}_g} \mathcal{L}^G(\boldsymbol{\theta}_d, \boldsymbol{\theta}_g)$;
end

As shown in Algorithm 1, the generative adversarial networks are simultaneously trained by updating the discriminative distribution in order that it differentiates data sampled from the real data distribution $p_{data}(\boldsymbol{S})$ from those sampled from the generative distribution $p_g(\hat{\boldsymbol{S}})$.

During adversarial training, the weight updating processes between G and D alternates. Empirically, after the parameters of D have been updated k times we update the parameters of G once. The performance of D and G improves progressively by this adversarial optimization procedure and the global optimal solution can be achieved if $p_{data} = p_g$. Ultimately, if the differentiation competence of D has been upgraded to a high level while unable to differentiate the p_g from p_{data} accurately, it is of the opinion that the generator G has learnt the distribution of the training data and we say the training converge.

C. RUL Prediction

After training, a present-to-end prediction could be made when given on-line measurements. Specifically, the generator takes the transformed feature data $\boldsymbol{S}_t$ as input at time t, and outputs $\boldsymbol{y}_t$ as the mixed parameters of the probability distributions of $\boldsymbol{S}_{t+1}$. By sampling from these generated Gaussian mixture model parameters and a categorical distribution we obtain the prediction of $\hat{\boldsymbol{S}}_{t+1}$. Different from the training process, the yielded prediction $\hat{\boldsymbol{S}}_{t+1}$ at time step t is reused as the input at time step $t+1$. When the health state indicator $\mathcal{HI}$ changes to $(0, 1)$, the prediction process is stopped and maintenance actions are required. The whole process of the on-line present-to-end prediction is presented in Algorithm 2, where t_p is the present time step and T_s is the prediction horizon.

The procedure of RUL prediction is based on the present-to-end prediction. Starting from the prediction time t_p, the algorithm calculates the prediction until the pair of health state indicators $\mathcal{HI} = (0, 1)$ is obtained at time step $\hat{T}_s$. Consequently, the predicted remaining useful life is defined as:

$$RUL = \hat{T}_s - t_p \tag{14}$$

where $t = 1, 2, \cdots, \hat{T}_s$.

Algorithm 2: Present-to-end degradation prediction.

Input: feature series $\boldsymbol{S} = \boldsymbol{S}_t,\ t = 1, 2, \cdots, t_p$
initialize $h_0 = 0$, $f_0 = 0$, $\mathcal{HI} = (1, 0)$;
initialize $\boldsymbol{S}_0 = (f_0, \mathcal{HI})$, $t = 0$;
while $\mathcal{HI} \,!= (0, 1)$ **do**
  Generate $\boldsymbol{h}_{t+1}$ and $\boldsymbol{y}_{t+1}$ by $\boldsymbol{h}_t$ and $\boldsymbol{S}_t$;
  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
Output: return $\boldsymbol{S} = \hat{\boldsymbol{S}}_t,\ t = t_p, \cdots, T_s$

IV. Experiments

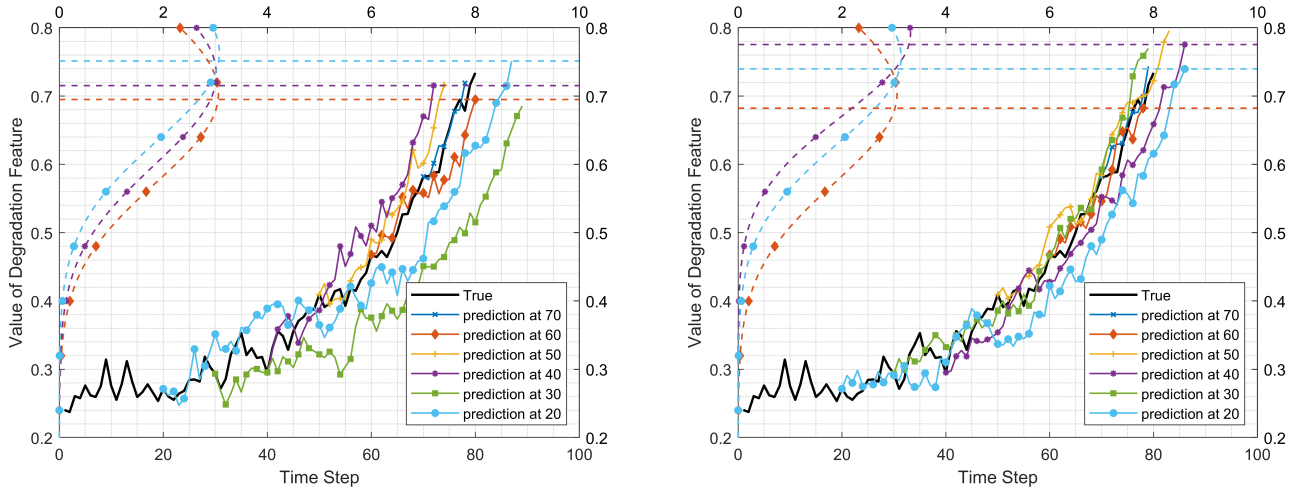
A. Experimental Setup

Model Layout Details: The LSTM network in generator consists of 256 internal (hidden) units. We used $M = 10$ components for the Gaussian mixture model. D has a bidirectional layout, whereas G is unidirectional.

Dataset Description: Datasets were created from the Modular Aero-Propulsion System Simulation (MAPSS), composed of multiple multi-variate time-series (24 sensors including 3 operational settings) from a fleet of aeroengines with dissimilar levels of initial wear and unspecified manufacturing disparity. The data contain 100 such engines and their associated run-to-failure time-series trajectories. However, part of the sensor observations have constant values across the engine's life cycle and do not contribute useful information for RUL prediction. For the sake of extracting the optimal feature sub-set, criteria coefficients of selected features are assessed by correlative and monotonic metrics [18]. Specifically, we observed that the $11th$, $12th$ and $13th$ sensor features have similar performance, thus we choose the $11th$ sensor feature as the input, and down sample all the $11th$ feature series to make the length of the each sequences (T_s) within 100 (T_{max}). At each time step $t < T_s$, we concatenated the health state indicator $\mathcal{HI} = (1, 0)$ to f_t and $\mathcal{HI} = (0, 1)$ to $T_s \leq t \leq T_{max}$.

Baseline Model: The baseline is a LSTM network similar to standalone G, trained entirely to predict the next status at each time step in the recurrence.

Implementation: Back propagation through time (BPTT) was employed, and the iteration steps fragment k of mini-batch stochastic gradient descent (SGD) was set to 4, with the batch size set to 10. The G was pre-trained for 10 epochs with loss



(a) Baseline Model

(b) LSTM-GAN Model

Fig. 2. Visualization of the present-to-end prediction result conducted at different time steps and the PDF estimation for the last feature generated by the GMM layer for both the baseline model and GAN model. The baseline model's prediction is quite random, while the GAN model shows the similar distribution of real degradation. In these figures, the prediction time steps are chosen at 20, 30, 40, 50, 60, and 70.

function $\mathcal{L}^G = \mathcal{L}_f^G + \mathcal{L}_h^G$. Layer normalization and recurrent dropout with a keep probability of 90% were applied. The learning rate was set to 0.01 and a gradient clipping of 1.0 was used. The implementation is built based on the publicly available Tensorflow platform. The whole framework is trained on NVIDIA Geforce GTX 1080 Ti and Titan Xp GPU with 32 GB memory.

Evaluation: Evaluation of the proposed model was done using the mean absolute error (MAE) of generated output as:

$$MAE_{degradation} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{T_s}\sum_{t=t_p}^{T_s}\left|\hat{\boldsymbol{S}}_t - \boldsymbol{S}_t\right|\right) \tag{15}$$

$$MAE_{RUL} = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{RUL}_t - RUL_t\right| \tag{16}$$

where T_s is the length of the selected test degradation series, $\hat{\boldsymbol{S}}_t$ is the predicted value, $\boldsymbol{S}_t$ is the true degradation value for comparison and N is the number of selected series.

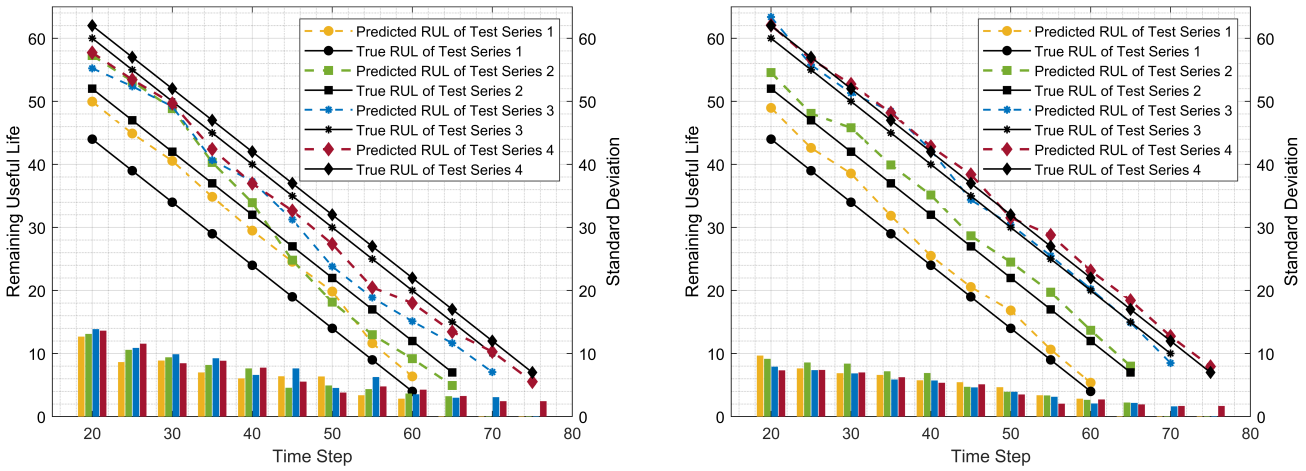
B. Results and Discussion

The present-to-end prediction results generated by the baseline and proposed LSTM-GAN models are shown in Fig. 2. The $MAE_{degradation}$ for all test series are listed in TABLE I. We visualized the prediction results of one randomly picked test series at every 10 time steps as an example. As shown in Fig. 2(a), the predicted degradation curves generated by the baseline model roughly represent the real trend, and the prediction becomes more accurate in the latter half of lifetime than the first half. As more observations are available, the result can be more accurate. As shown in Fig. 2(b), the LSTM-GAN is able to generate more accurate degradation curves, which indicates the adversarial training could help LSTM learn the distribution of real data better. Compared to the baseline model, according to $MAE_{degradation}$ in TABLE I, the LSTM-GAN makes accurate degradation progress prediction even at the early stage.

Fig. 3 presents the RUL prediction from four randomly picked test series when compared with corresponding real RUL value. The RUL prediction is visualized at every 5 time steps. Compared to the baseline model, the RUL prediction estimated by LSTM-GAN model is more accurate. Besides predicting RUL, we also evaluate the standard deviation of each prediction using a histogram. For both models, earlier prediction leads to larger standard deviations. This is because the predicted $\hat{\boldsymbol{S}}_t$ is sampled from GMM and categorical distribution rather than generated directly. Another reason is that RUL is estimated based on longer observed sequence information in latter part of the lifetime and it is easier to predict the adjoining forthcoming than the long-horizon. To quantify the observations, the MAE_{RUL} over all the test series are listed in TABLE I, which again verified the effectiveness of our proposed approach compared with the baseline model in both degradation progress and RUL prediction tasks.

V. Conclusions

In this paper, we proposed an adversarial learning approach for prediction of the long-term degradation progress and remaining useful life without the need of any prior failure threshold knowledge. We used correlative and monotonic metrics to identify critical features in the degradation progress, which were then concatenated with health state indicator to construct the training dataset. The proposed approach employed LSTM as generative model by fully exploiting its potentiality of learning long-term dependencies in time-series data. Through a fully-connected layer, the output of the LSTM was projected into the parameters of a Gaussian mixture model and a categorical distribution for sampling consequent predictions. Experiments on MAPSS data of aeroengines verified



(a) Baseline Model

(b) LSTM-GAN Model

Fig. 3. Visualization of the RUL prediction result conducted at different time step (i.e., at every 5 time steps from 20 to 70) for both the baseline model and the LSTM-GAN model. The histogram at the bottom shows the standard deviation of 50 times repeated RUL prediction at each compared time step.

TABLE I
PERFORMANCE COMPARISON.

Time Steps	Baseline		Adversarial Training	
	$MAE_{degradation}$	MAE_{RUL}	$MAE_{degradation}$	MAE_{RUL}
20	0.0466	5.07	0.0476	2.74
25	0.1034	4.57	0.0446	1.44
30	0.0872	4.13	0.0229	2.60
35	0.0872	4.52	0.0219	2.59
40	0.0457	3.96	0.0342	1.93
45	0.0815	3.79	0.0241	1.31
50	0.0329	4.13	0.0266	1.56
55	0.0346	3.82	0.0148	1.67
60	0.0305	3.51	0.0183	1.15
65	0.0248	2.98	0.0115	0.84
70	0.0132	1.70	0.0177	0.86

the effectiveness of the proposed framework. The adversarial training enables the LSTM to better capturing the distribution of real degradation progress, thus leading to a more accurate RUL prediction.

In the future, more cost-effective sensitive feature extraction method, such as Graph convolution networks (GCN) based technique, will be studied. Follow-up experiments on multi-faults situations and more intricate data, such as the non-stationary wind turbine data [20], will be carried out to verify the robustness of the proposed approach.

ACKNOWLEDGMENT

This work was supported by the National Science Foundation under grant ECCS-1809164 and the Key Research and Development Project of Hunan Province, China under grant no. 2018WK2022.

REFERENCES

[1] Y. Peng, M. Dong, and M. J. Zuo, "Current status of machine prognostics in condition-based maintenance: a review," *The International Journal of Advanced Manufacturing Technology*, vol. 50, no. 1, pp. 297–313, Sep 2010. [Online]. Available: https://doi.org/10.1007/s00170-009-2482-0

[2] J. Shimada and S. Sakajo, "A statistical approach to reduce failure facilities based on predictive maintenance," in *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, 2016, pp. 5156–5160.

[3] 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.

[4] X. Li, Q. Ding, and J.-Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Reliability Engineering & System Safety*, vol. 172, pp. 1–11, 2018.

[5] Z. Tian, "An artificial neural network method for remaining useful life prediction of equipment subject to condition monitoring," *Journal of Intelligent Manufacturing*, vol. 23, no. 2, pp. 227–237, 2012.

[6] B. Y. Bejarbaneh, E. Y. Bejarbaneh, A. Fahimifar, D. J. Armaghani *et al.*, "Intelligent modelling of sandstone deformation behaviour using fuzzy logic and neural network systems," *Bulletin of Engineering Geology and the Environment*, vol. 77, no. 1, pp. 345–361, 2018.

[7] P. Lim, C. K. Goh, and K. C. Tan, "A time window neural network based framework for remaining useful life estimation," in *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, 2016, pp. 1746–1753.

[8] 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.

[9] F. O. Heimes, "Recurrent neural networks for remaining useful life estimation," in *Prognostics and Health Management, 2008. PHM 2008. International Conference on*. IEEE, 2008, pp. 1–6.

[10] A. S. Yoon, T. Lee, Y. Lim, D. Jung, P. Kang, D. Kim, K. Park, and Y. Choi, "Semi-supervised learning with deep generative models for asset failure prediction," *arXiv preprint arXiv:1709.00845*, 2017.

[11] J. Li, S. Liu, H. He, and L. Li, "A novel framework for gear safety factor prediction," *IEEE Transactions on Industrial Informatics*, 2018.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[13] Y. Ban, J. Pu, Y. Chen, and Y. Wang, "NEGAN: Network embedding based on generative adversarial networks," in *2018 International Joint Conference on Neural Networks (IJCNN)*, July 2018, pp. 1–8.

[14] Z. Zhai and J. Zhai, "Identity-preserving conditional generative adversarial network," in *2018 International Joint Conference on Neural Networks (IJCNN)*, July 2018, pp. 1–5.

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

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[18] 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.

[19] O. Mogren, "C-rnn-gan: Continuous recurrent neural networks with adversarial training," *arXiv preprint arXiv:1611.09904*, 2016.

[20] G. Jiang, H. He, P. Xie, and Y. Tang, "Stacked multilevel-denoising autoencoders: A new representation learning approach for wind turbine gearbox fault diagnosis," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 9, pp. 2391–2402, Sep. 2017.