2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)
August 23-27, 2021. Lyon, France

Time-Gated Remaining Useful Life Prediction with Non-Periodical Inspection Data

Yingjun Deng[1], Mingpeiyu Zhang[2], Tian Wang[3], and Huaming Wu[1]

2021 IEEE 17th International Conference on Automation Science and Engineering (CASE) | 978-1-6654-1873-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/CASE49439.2021.9551607

***Abstract*— This paper addresses the problem of remaining useful lifetime (RUL) prediction with non-periodical inspection data. To construct the RUL predictor, a two-stage solution is presented with the recently proposed time-gated long short-term memory network (TGLSTM) and a surrogate Wiener propagation model. The TGLSTM is used to process the non-periodical time-stamps and the Wiener propagation model aims to control the sequence-wise uncertainty. Also to process the high-frequency sensory data during each inspection, the temporal convolutional network (TCN) is introduced. The modeling rationale comes from the observed fact that prediction uncertainty reduces when time tends to the failure time, and the key insight is to introduce the latent Wiener process to model the joint probability density to observe the RUL prediction from the TGLSTM predictor and the actual RUL record simultaneously. Moreover, the TGLSTM predictor is interactively trained with the uncertainty propagation model. Our model is validated using the non-periodically under-sampled data from a turbofan engine degradation simulation use case.**

I. INTRODUCTION

In modern industrial systems, remaining useful lifetime (RUL) prediction usually refers to system condition data that may come from sensors or human-aided inspections in the field [1], [2]. The RUL prediction is also a widely-adopted risk indicator for maintenance planning, whose uncertainty is closely related to the real-time evaluation of system reliability and predictive maintenance policies [3].

This paper addresses three challenges for RUL prediction induced by the high-dimensionality, non-periodicity, and uncertainty of inspection data.

- *Feature extraction from high-frequency sensory data*: the inspection data is usually a piece of high-frequency sequence from sensors. For prediction or decision needs, directly using the high-frequency data is difficult such that feature extraction or dimension reduction are needed for RUL prediction. This paper refers to the temporal convolutional network [4] to solve this challenge.
- *Information processing for non-periodical inspections*: the inspection intervals may not be equidistant due to signal delay, unscheduled maintenance, unsynchronized system time etc. This leads to the difficulty to maximize the use of variable-size, non-periodical inspection data, and to keep data balance during predictor training simultaneously. For non-periodical scalar data, this usually can be partially solved by degradation process based estimation with the Markov hypothesis [2], [5]. However facing multi-dimensional data, non-periodical data makes big trouble for structuring the input data for popular machine learning models [6]. A recently proposed time-gated long short-term memory (TGLSTM) network [7] will be considered to solve this challenge.
- *Uncertainty control in sequential prediction*: the inspection data are noisy and show variety due to different types of uncertainty, e.g. aleatory and epistemic. The system uncertainty is propagated to the observation uncertainty and further RUL prediction uncertainty [8]. Especially when considering the RUL prediction sequentially, the sequence-wise uncertainty is not fully understood and well-controlled by normal point-wise loss functions, e.g. mean square error (MSE), if the RUL predictor is trained in a supervised learning way. To quantify and control the sequence-wise RUL prediction uncertainty, the Wiener propagation control model for periodical inspection data [9] will be modified for non-periodical scenarios in this paper.

Conventional statistical prediction models like time series focus on periodic scenarios [5], and this paper considers a hybrid-model for non-periodic data that consists of two stages: 1) the stacked model of TCN and TGLSTM [7] will be adopted to construct a RUL predictor, due to its self-learning capacity for the high-dimensional data in non-periodical inspection scenarios; 2) the drifted Wiener process will be considered as a surrogate model to quantify and control the uncertainty propagation for RUL prediction.

The hybrid model between machine learning and stochastic modeling contributes to uncertainty quantification while the interpretability and transparency are missing for black-box-like machine learning models [9], [10]. This is motivated by the current trend that many machine learning models, especially deep neural networks, aims at better mapping between inspection data and RUL records with a focus on testing accuracy. However RUL prediction and associated uncertainty are related to all pending future inspection data before the system failure, and the direct inspection-RUL mapping does not reveal how the inspection data is projected to RUL prediction. This paper proposed a two-stage

This work is partially supported by National Natural Science Foundation of China (71701143, 61972016), Science and Technology on Particle Transport and Seperation Laboratory, Beijing Natural Science Foundation (L191007) and Tianjin AI Key Project (19ZXZNGX00050) .

[1]Yingjun Deng and Huaming Wu are with Center for Applied Mathematics, Tianjin University, Tianjin 300072, China.{yingjun.deng,whming}@tju.edu.cn

[2]Mingpeiyu Zhang is with Microsoft Search Technology Center Asia, Suzhou, China. mingpeiyu.zhang@microsoft.com

[3]Tian Wang is with Institute of Artificial Intelligence, Beihang University, Beijing 100191, China. tianwang@buua.edu.cn

model for RUL prediction, such that the RUL prediction is processed essentially as a health indicator and the system failure is modeled by first passage time.

Specifically, the TGLSTM will be introduced as a RUL predictor, and the output is processed as observation to a Wiener [11] propagation model. The Wiener propagation model bridges between the feature extracted from high-dimensional sensor readings and scalar RUL predictions. Compared with the hidden discrete-state space in classical hidden Markov models, our work is more flexible and applicable since a continuous-state hidden space is considered.

Our contribution is threefold.

- A solution for the real-time RUL prediction with non-periodical, high-frequency inspection data is proposed.
- The non-periodical inspection data is processed by the stacked model of TCN [4] and TGLSTM [7].
- A surrogate Wiener propagation model is introduced to control the trade-off between prediction accuracy and uncertainty.

The paper is organized as follows. Section II will introduce three related research topics. In Section IV, the solution for RUL prediction with non-periodical inspection data will be discussed, and the hybrid modeling framework based on the TGLSTM and the surrogate Wiener propagation will be presented. Two real-world case studies will be presented in Section V. Conclusions are made at the end.

II. Related Work

A. Temporal Convolutional Network for Remaining Useful Lifetime Prediction

The conventional recurrent models, e.g. LSTM and GRU are almost standard choices for sequence modeling, and RUL prediction [9], [12], [13]. However, these recurrent models can hardly process extremely long sequences even different memory mechanisms are introduced, say a sequence of 250k data points per second. Also as the information is processed sequentially over time in recurrent models, the computation can hardly be parallel and accelerated by GPU.

The temporal convolution network (TCN) [4], [14], [15] is a promising solution to process high-frequency sensory data using dilated causal convolution and 1D convolution. Using different hidden layers for perception of different time horizons, TCN is also much faster than the recurrent model. Previous trials are more related to acoustic studies, e.g. WaveNet [16], but later extend to more general multivariate time series [17], [18]. For the task of remaining useful lifetime prediction [5], the input is commonly time series from different sensors which is essentially the same with acoustic signals. TCN gets more and more attention from the reliability engineering society recently [19], [20].

B. Time-Gated LSTM for non-periodical time series

The inspection operation may not be periodically scheduled so that the time intervals in an inspection sequence are not uniform. Conventional sequence models like RNN, LSTM, GRU [12] and also the aforementioned TCN [4] are not applied in such a scenario due to the strong assumption of equidistant time intervals. To process the non-uniform time intervals, the time-gate is introduced into the normal LSTM [7], such that the prediction becomes time-dependent.

C. Sequence-wise Uncertainty Propagation Control

Note that the RUL prediction task [21] using machine learning tools commonly considers the point-wise prediction. However, the degradation modeling using stochastic processes [5] is usually considered for a scalar time series. The degradation modeling aims to reveal a latent degradation process to represent the system state transition, which is different from the point-wise prediction accuracy. To consider the sequence-wise prediction accuracy, a surrogate Wiener propagation model was introduced in [9], which accepts the blackbox predictor' output as observation to the latent drifted Wiener process.

III. Problem Statement

Throughout this paper, the following scenario is considered for an unspecified industrial system.

- The industrial system fails at time $\tau \geq 0$ which cannot be observed until the failure happens.
- Inspections can possibly be taken on the system at time $s \in [0, \tau)$, which return a multi-dimensional vector $\mathbf{x_s} \in \mathbb{R}^m$.
- Due to the non-periodical inspection policy, the inspection data may not come periodically. Hence $s_j, j = 1, \cdots, n$ is denoted as the j-th inspection time, with $\tau = s_n$ as the last inspection time. At time $s = s_j$, all previous inspection data (including those at time s), $\{\mathbf{x_{s_k}}, k = 1, \cdots, j\}$ are represented by $\mathbf{x_{0:s_j}}$, or $\mathbf{x_{0:s}}$.
- In the historical data for run-to-failure tests, τ is observed and at time $s \in [0, \tau)$, the RUL value equals $r_s := \tau - s$.

The aim of RUL prediction is to find a map φ from all previous inspection data $\mathbf{x_{0:s_j}} \in \mathbb{R}^{\mathbf{m}\times\mathbf{j}}$ into the estimate of RUL $r_s \in \mathbb{R}^+$, say α_{s_j},

$$\varphi : \mathbb{R}^{m\times j} \longrightarrow \mathbb{R}^+, \quad \text{with} \quad \varphi(\mathbf{x_{0:s_j}}) = \alpha_{s_j}. \tag{1}$$

In the periodical inspection scenarios, the RUL prediction can be done based on normal recurrent models, e.g. recurrent neural networks and related variants like LSTM networks [12]. However, in non-periodical inspection scenarios, normal recurrent models cannot be applied directly since the iterative relationship does not hold automatically. The dependence of two consecutive inspections depends largely on the time interval which cannot be ignored in the prediction model. To process the non-periodical sequences, in this paper we consider a time-gated recurrent model [7] formulated as follows,

$$\varphi(\mathbf{x_{0:s_j}}) = \begin{cases} h_0 = \theta_0; \\ h_k = \sigma_h(h_{k-1}, \mathbf{x_{s_k}}, \Delta s_k; \theta_h), k = 1, \cdots, j; \\ \alpha_{s_j} = \sigma_a(h_j), \end{cases} \tag{2}$$

where $\Delta s_k = s_k - s_{k-1}$ is the inspection interval, σ_h, σ_a are real-valued (vector) functions and θ_0, θ_h are trainable

vector parameters. It is noted that $\Delta s_k \equiv 1$ holds in normal recurrent models by default. The time-gated recurrent model (2) will be specified as TGLSTM in the next sub-section.

IV. Sequential RUL Prediction with Time-Gated Neural Networks

A. Time-Gated LSTM Network

Theoretically, in the input sequences, classic (or "vanilla") RNNs can keep track of arbitrary long-term dependence. The issue with vanilla RNNs is computational (or practical) in nature: when training a vanilla RNN using back-propagation, the gradients which are back-propagated can "vanish" (that is, they can tend to zero) or "explode" (that is, they can tend to infinity), due to the computations involved in the process, which use finite-precision numbers. RNNs using LSTM units partially solve the vanishing gradient problem, because LSTM units allow gradients to also flow unchanged.

The conventional LSTM can be formulated as follows [12], [7],

$$i_t = \sigma(W_i x_t + R_i h_{t-1} + b_i) \tag{3}$$
$$f_t = \sigma(W_f x_t + R_f h_{t-1} + b_f) \tag{4}$$
$$o_t = \sigma(W_o x_t + R_o h_{t-1} + b_o) \tag{5}$$
$$\tilde{c}_t = \sigma_g(W_z x_t + R_z h_{t-1} + b_c) \tag{6}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{7}$$
$$h_t = o_t \odot \sigma_h(c_t) \tag{8}$$

where the initial values are $h_0 = 0$ and $c_0 = 0$, $\odot$ is the element-wise (Hadamard) product and operates on the two vectors of the same size. The subscript t denotes the time step. $x_t \in \mathbb{R}^m$ is the input vector to the LSTM unit, $c_t \in \mathbb{R}^h$ is the cell state vector and $h_t \in \mathbb{R}^h$ is the hidden state vector also the output vector at time t. $\tilde{c}_t \in \mathbb{R}^h$ is the cell input activation vector, and i_t , f_t, $o_t \in \mathbb{R}^h$ are the input, forget and output gates' activation vectors. $W \in \mathbb{R}^{h \times m}$ are the input weight matrices, $b \in \mathbb{R}^h$ are the bias metrics, and $R \in \mathbb{R}^{h \times h}$ are the recurrent weight matrices. These three sets of parameters are being learned during training. $\sigma_g(\cdot)$, $\sigma_h(\cdot)$ and $\sigma(\cdot)$ are nonlinear activation functions, which apply the point-wise operations. Hyperbolic tangent function $\tanh(\cdot)$ is usually used for $g(\cdot)$ and $h(\cdot)$ functions and $\sigma(\cdot)$ is the sigmoid function.

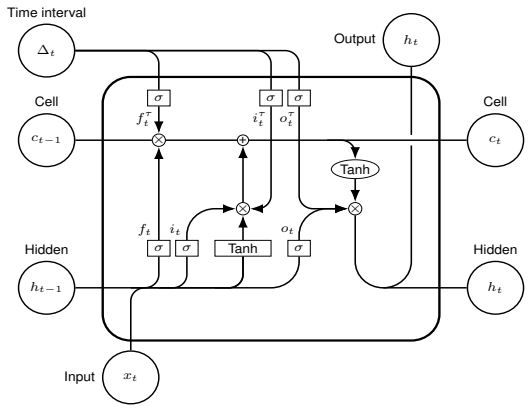


Fig. 1: Time-gated LSTM structure.

To introduce the time-dependence into the prediction model as explained in Eq. (2), TGLSTM [7] introduced three different time gates which incorporate the time information as a nonlinear scaling factor on the conventional gates of LSTM. It enables the original LSTM gates to give different responses depending on the time intervals. A nonlinear activation function $\sigma_\tau(\cdot)$ is used to model this scaling effect. In addition to (7)-(8), the forward-pass process of the new LSTM architecture in Fig. 1 is modeled by the following equations [7]:

$$i_t^\tau = \sigma_\tau(W_i^\tau \Delta t + b_i^\tau) \tag{9}$$
$$f_t^\tau = \sigma_\tau(W_f^\tau \Delta t + b_f^\tau) \tag{10}$$
$$o_t^\tau = \sigma_\tau(W_o^\tau \Delta t + b_o^\tau) \tag{11}$$
$$c_t = f_t \odot c_{t-1} \odot f_t^\tau + i_t \odot \tilde{c}_t \odot i_t^\tau \tag{12}$$
$$h_t = o_t \odot \sigma_h(c_t) \odot o_t^\tau \tag{13}$$

where W_i^τ, W_f^τ, W_o^τ are the weight matrices of the time gates, σ_τ is the point-wise non-linearity, which is set to the sigmoid function $\sigma(\cdot)$. $\Delta t \in \mathbb{R}$ is the time intervals for the input sequence at time t, as the input for the time gates. The overall TGLSTM unit is illustrated in Fig. 1.

B. A Two-layer Stacked Model using Time-Gated LSTM and Temporal Convolution Network

To promise the high-fidelity of data acquisition from a modern industrial system, the sampling rate is usually set to be a very high frequency; and each inspection operation returns a piece of high-frequency time series. For instance, by Nyquist–Shannon sampling theorem [22], the sampling rate must be at least two times of the rolling frequency 20kHz of a rolling machine, that is 240k data points per minute. Due to the data redundancy and the limited computing capacity, the inspection operation is commonly realized by taking the data points in a small piece of time, say 1 second.

In such a scenario, TGLSTM can be used to process the coarse-grained information refined from the inspection. However, to incorporate the high-frequency data with the TGLSTM, the temporal convolutional network (TCN) [4] will be considered to process the piece of high-frequency data in each inspection.

As illustrated in Fig. 2, the network consists of two parts, the TCN part and the TGLSTM part. The TCN takes the high frequency data as input, extract and encode the feature to a lower dimension. We use the same dilated Convolutions structure as described in [4], with dilation factors $d = 1, 2, 4$ and filter size $k = 3$.

C. Sequence-wise Uncertainty Propagation Control

As the RUL prediction updates over time for a uniform failure time during the system life-cycle, so the prediction uncertainty should decrease as new information comes. Following the previous work in [9], we will use a surrogate Wiener propagation model to control the sequence-wise uncertainty in RUL prediction. Specifically for $t > s$, given

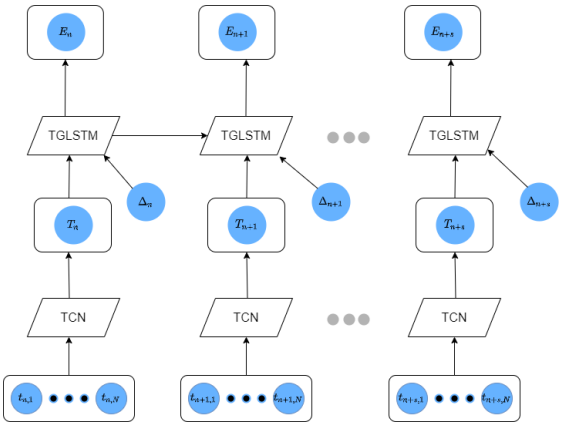


Fig. 2: Stacked TCN and TGLSTM network.

two predictions at time t and s respectively, say α_t and α_s, the prediction α_t is assumed to be an observation to the following drifted Wiener process[11], [2],

$$Y_t = -t + \frac{W_t}{\sqrt{c}}, t \geq s, \tag{14}$$

with $Y_s = \alpha_s$ as the initial prediction and W_t is a standard Wiener process. $c > 0$ measures the uncertainty propagation rate.

Note that the surrogate modeling for periodical inspections in [9] can be reproduced in the non-periodical inspection scenario. Under the assumption $\alpha_\tau = 0$ and from the time-reversal property of Wiener process, α_s follows the normal distribution with mean RUL r_s and variance r_s/c [9]. Moreover with the latent process Y_t, the RUL prediction at time s is modeled as a first passage time conditional on the observed value α_s, i.e. a random variable given by

$$R_s := \inf_{t \geq 0}\{t : Y_{t+s} \leq 0 | Y_s = \alpha_s, \tau > s\}. \tag{15}$$

As the first passage time of a Wiener process follows the inverse Gaussian distribution [11], $R_s \sim \mathcal{G}(\alpha_s, c)$, where $\mathcal{G}(\alpha, c)$ represents the inverse Gaussian distribution with the probability density function defined by

$$\mathcal{G}(x; \alpha, c) = \left[\frac{c\alpha^2}{2\pi x^3}\right]^{1/2} \exp\left\{\frac{-c(x-\alpha)^2}{2x}\right\}, x > 0. \tag{16}$$

Note that if $X \sim \mathcal{G}(\alpha, c)$, then $\mathbb{E}(X) = \alpha, \mathrm{Var}(X) = \alpha/c$.

At time $s = s_j$ (i.e. the j-th inspection time), given the prediction sequence $\alpha_{0:s_j}$ and the RUL observation r_{s_j}, the prediction loss is established by the negative log-likelihood function to observe both sequences simultaneously,

$$\ell(r_{0:s_j}, \alpha_{0:s_j}) = -2\Bigg[\log p_{Y_{s_1}}(\alpha_{s_1}) + \log p_{R_{s_j}}(r_{s_j}|Y_{s_j} = \alpha_{s_j}) + \sum_{i=1}^{j} \log p_{Y_{s_i}}(\alpha_{s_i}|Y_{s_{i-1}} = \alpha_{s_{i-1}})\Bigg], \tag{17}$$

where $p_Y(y), p_Y(y|X = x)$ represent the probability density and conditional density function, respectively. It is noted that (17) depends on the predictor's parameter and the latent process parameter.

D. Model Training

The joint likelihood hood (17) provides a pair-wise prediction loss to evaluate the RUL predictor. A batch gradient descent algorithm naturally comes to train the predictor using existing inspection-RUL pairs. Moreover, the available training data $\mathcal{D}_{train}$ is specified as follows.

- Inspection data from total N run-to-failure tests are available for training.
- For the i-th test, totally n_i inspections are taken at times $s_{ij}, j = 1, ..., n_i$; the failure happens at τ_i such that at s_{ij} correspondingly the RUL equals $r_{ij} = \tau_i - s_{ij}$; the inspection data at s_{ij} is denoted as $\mathbf{x}_{\mathbf{s_{ij}}}$.

Denote the trainable vector parameter for the RUL predictor given by (1) by θ, and $\alpha_{s_{ij}}$ denotes the prediction $\alpha_{s_{ij}} = \varphi(\mathbf{x}_{\mathbf{0:s_{ij}}}; \theta)$. Combining with the uncertainty propagation rate c in (14), the prediction loss naturally comes as the likelihood for all inspection-RUL pairs from (17),

$$\mathcal{L}_p(\theta, c; \mathcal{D}_{train}) = \sum_{i=1}^{N} \ell(\alpha_{0:s_{ij}}, r_{ij}). \tag{18}$$

Hence the model training is to find the solution to the following joint optimization problem,

$$(\theta^*, c^*) \in \arg\min_{\theta, c} \mathcal{L}_p(\theta, c; \mathcal{D}_{train}). \tag{19}$$

Due to the explicit derivative of $\mathcal{L}_p$ w.r.t. c, the optimization for c can be realized by letting the corresponding derivative be zero [9]. Hence, the model training can be done with an alternating direction algorithm that is summarized in Algorithm 1. For practical convenience, we let the gradient descent goes for M epochs, and then select the optimal model that reaches the minimal loss during training.

Algorithm 1 Alternating direction optimization

Require:
1) Training data $\mathcal{D}_{train} = \{(\mathbf{x}_{\mathbf{0:s_{in_i}}}, \tau_i)\}_{i=1}^N$;
2) Maximum number of training epochs $M \in \mathbb{N}$.

Ensure: The RUL predictor $\varphi(\cdot; \theta^*)$ and the uncertainty propagation rate c^* to minimize the loss $\mathcal{L}_p(\theta, c; \mathcal{D}_{train})$.

1: Initiate $i = 1, c_1, \theta_1$;
2: **for** $i \leq M$ **do**
3: Solve c_{i+1} from
$$\frac{\partial \mathcal{L}_p(\theta_i, c; \mathcal{D}_{train})}{\partial c} = 0;$$
4: Update θ_{i+1} for θ by gradient-descent algorithms using the loss function $L_p(\theta, c_{i+1}; \mathcal{D}_{train})$;
5: $i + 1 \to i$;
6: **end for**
7: $idx = \arg\min_{i=1,\cdots,M} L_p(\theta_i, c_i; \mathcal{D}_{train})$;
8: **return** $\theta_{idx} \to \theta^*$, $c_{idx} \to c^*$.

V. Empirical Studies

A. NASA CMAPSS Turbofan Engine Dataset

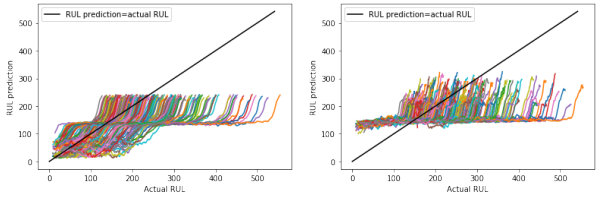
In this section, the performance of the hybrid model of TGLSTM and Wiener propagation will be presented on the

CMAPSS Turbofan Engine Degradation Simulation Datasets [13]. The data was from simulation of engine degradation using CMAPSS[13].

Under distinct combinations of operational circumstances and fault modes, four distinct sets were simulated. The training data consists of multiple multivariate time series with "cycle" as the time unit, together with 21 sensor readings for each cycle. Each time series can be assumed as being generated from a different engine of the same type. The testing data has the same data schema as the training data. The only difference is that the data does not indicate when the failure occurs. Finally, the ground truth data provides the number of remaining working cycles for the engines in the testing data. Throughout this paper, those datasets labeled by $FD004$ will be considered.

Note that the CMAPSS datasets are constantly recorded. To make the tests more indicative for non-periodical inspection scenarios, the training data-sets are randomly sub-sampled such that 249 sub-sequence of size 50 with non-equidistant time-intervals are extracted from training sequences. Furthermore, the TGLSTM predictor consists of 256 hidden units with a ReLu output layer, and the optimizer is selected as the RMSprop with an initial learning rate of 0.001. All tests are done in a workstation with AMD 2950X and Nvidia GTX 1080Ti, which are coded with Keras [23] and Tensorflow [24].

We trained and compared the TGLSTM predictor with $\mathcal{L}_p$-loss (18) and mean absolute percentage error (MAPE) for 10000 epochs. Trained models are called Wiener-hidden and MAPE-oriented predictor respectively. The sequence-wise test performance is illustrated in Fig. 3.



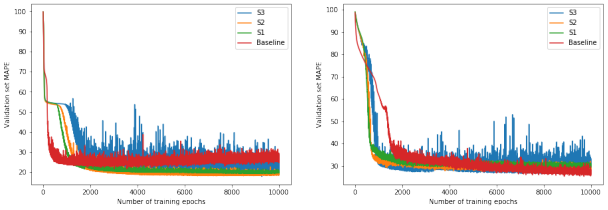
(a) Wiener-hidden prediction (b) MAPE-oriented prediction

Fig. 3: Sequence-wise prediction for the Wiener-hidden and MAPE-oriented TGLSTM over actual RUL for test data.

To validate the impact of the time interval between samples on the model, we use different sample rates to generate non-uniformly sampled data. The sequence length is fixed to 20. For each sequence, the instances are drawn randomly without replacement from every 80, 100 and 120 instances in the training set for S1, S2 and S3 simulations. So the expectation of the time intervals for these three setups are 4, 5 and 6 respectively, and the corresponding sample rates are 0.25, 0.20 and 0.17. We used the set without under-sampling as baseline data set. The TGLSTM predictor consists of 100 hidden units with a ReLu output layer, and the optimizer is selected as the SGD with an initial learning rate of 0.001. And the model is trained with $\mathcal{L}_p$-loss (18) and mean absolute percentage error (MAPE) for 10000 epochs.

TABLE I: Results comparison between models with different loss on different under-sampling data sets

Sampling set	Training Loss	MAPE
Baseline	MAPE loss	27.08
	$\mathcal{L}_p$-loss	27.59
S1	MAPE loss	19.95
	$\mathcal{L}_p$-loss	30.28
S2	MAPE loss	19.19
	$\mathcal{L}_p$-loss	29.21
S3	MAPE loss	22.29
	$\mathcal{L}_p$-loss	29.62



(a) Trained with MAPE loss. (b) Trained with $\mathcal{L}_p$-loss

Fig. 4: MAPE for different sample rates on Validation Dataset

Fig. 4a shows the MAPE on the testing set during training for different sample rates with MAPE loss. And Table I shows the MAPE on the testing set of models trained with different losses and on different sampled data sets at the end of the training. We can see that on all the sample rates, the model can ultimately get similar performance on the testing set. When using the original data set, the MAPE of models trained with $\mathcal{L}_p$-loss and MAPE loss can get about the same MAPE on the testing set. As the sample rate goes down, both MAPE loss based model and $\mathcal{L}_p$-loss based model keep about the same performance on the testing set.

VI. Conclusions and Discussions

This paper provided a solution to investigate RUL prediction uncertainty in high-dimensional and non-periodical inspection scenarios. In such scenarios, directly modeling uncertainty in inspection data faces the curse of dimension such that classical stochastic degradation models may not work. This paper adopted the surrogate modeling idea to establish the input-output evaluation criterion from historical RUL records, and the time-dependent uncertainty in RUL prediction is controlled by the surrogate Wiener propagation model.

Furthermore, the practical guide for hybrid modeling with machine learning and stochastic processes is presented. The hybrid modeling of aforementioned TGLSTM-TCN and Wiener propagation faces several optimization and modeling challenges. Our trial to connect parametric statistical models

and non-parametric machine learning in non-periodical inspection scenarios is relatively new. However, the empirical studies did not provide good prediction results based on the TGLSTM-TCN model. We are still working towards to figure out whether the non-periodic problem setting or the model setup leads to the bad performance.

REFERENCES

[1] J. Zhu, N. Chen, and W. Peng, "Estimation of bearing remaining useful life based on multiscale convolutional neural network," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 4, pp. 3208–3216, April 2019.

[2] Z. Zhang, X. Si, C. Hu, and Y. Lei, "Degradation data analysis and remaining useful life estimation: A review on wiener-process-based methods," *European Journal of Operational Research*, vol. 271, no. 3, pp. 775–796, 2018.

[3] S. Alaswad and Y. Xiang, "A review on condition-based maintenance optimization models for stochastically deteriorating system," *Reliab. Eng. Syst. Saf.*, vol. 157, pp. 54–63, jan 2017. [Online]. Available: http://dx.doi.org/10.1016/j.ress.2016.08.009 https://linkinghub.elsevier.com/retrieve/pii/S0951832016303714

[4] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018.

[5] X.-S. Si, W. Wang, C.-H. Hu, and D.-H. Zhou, "Remaining useful life estimation - a review on the statistical data driven approaches," *European Journal of Operational Research*, vol. 213, no. 1, pp. 1 –14, 2011.

[6] P. Wang, Y. Li, and C. K. Reddy, "Machine Learning for Survival Analysis," *ACM Comput. Surv.*, vol. 51, no. 6, pp. 1–36, feb 2019. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3303862.3214306

[7] S. O. Sahin and S. S. Kozat, "Nonuniformly Sampled Data Processing Using LSTM Networks," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 30, no. 5, pp. 1452–1461, may 2019.

[8] S. Sankararaman and K. Goebel, "Remaining useful life estimation in prognosis: An uncertainty propagation problem," *AIAA Infotech Aerosp. (I A) Conf.*, pp. 1–8, 2013.

[9] Y. Deng, A. D. Bucchianico, and M. Pechenizkiy, "Controlling the accuracy and uncertainty trade-off in RUL prediction with a surrogate Wiener propagation model," *Reliab. Eng. Syst. Saf.*, vol. 196, p. 106727, apr 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0951832019301723 https://linkinghub.elsevier.com/retrieve/pii/S0951832019301723

[10] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature*, vol. 521, no. 7553, pp. 452–459, 2015.

[11] M. Jeanblanc, M. Yor, and M. Chesney, *Mathematical methods for financial markets*. Springer Science & Business Media, 2009.

[12] A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer Berlin Heidelberg, 2012.

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

[14] Y. Zhao, Y. Xiong, and D. Lin, "Trajectory convolution for action recognition," in *Adv. Neural Inf. Process. Syst.*, 2018.

[15] K. Kang, W. Ouyang, H. Li, and X. Wang, "Object detection from video tubelets with convolutional neural networks," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2016.

[16] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[17] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *IJCAI Int. Jt. Conf. Artif. Intell.*, 2019.

[18] R. Wan, S. Mei, J. Wang, M. Liu, and F. Yang, "Multivariate temporal convolutional network: A deep neural networks approach for multivariate time series forecasting," *Electron.*, 2019.

[19] L. Ren, Y. Liu, X. Wang, J. Lü, and M. J. Deen, "Cloud-edge based lightweight temporal convolutional networks for remaining useful life prediction in iiot," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[20] L. Jayasinghe, T. Samarasinghe, C. Yuen, J. C. N. Low, and S. S. Ge, "Temporal convolutional memory networks for remaining useful life estimation of industrial machinery," 2018.

[21] X.-S. Si, W. Wang, C.-H. Hu, and D.-H. Zhou, "Estimating remaining useful life with three-source variability in degradation modeling," *Reliability, IEEE Transactions on*, vol. 63, no. 1, pp. 167–190, March 2014.

[22] C. E. Shannon, "Communication in the presence of noise," *Proc. IEEE*, 1998.

[23] F. Chollet *et al.*, "Keras," 2015. [Online]. Available: https://keras.io

[24] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/