Prognostics and Maintenance Decision-making for Mechanical Systems based on Condition Monitoring Data

by

Rui He

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

ENGINEERING MANAGEMENT

Department of Mechanical Engineering
University of Alberta

Abstract

Condition-based maintenance (CBM) is a maintenance approach that uses condition monitoring data to make maintenance decisions. The goal of CBM is to avoid machine shutdowns, reduce maintenance costs, and improve system safety. In modern mechanical systems, a wide range of sensors are used to collect data on the condition of different components. This has led to the proposals of various models, ranging from physics-based to data-driven methods, for predicting the remaining useful life (RUL) of components. Artificial intelligence techniques, specifically deep learning methods, have become increasingly popular in prognostics and health management due to their ability to process large amounts of data. The computational capabilities of machines have also advanced, allowing for numerical simulations that closely resemble real-world maintenance actions. These advancements provide more opportunities to utilize prognostic techniques and real-time data to support system maintenance.

However, in order to efficiently schedule maintenance using prognostic information, it is crucial to approach issues from both the component and system perspectives. Specifically, at the component level, it is necessary to develop prognostic techniques that can handle situations where there is a scarcity of data. In practice, engineering assets are often replaced before failures, resulting in a small number of failure histories being available for analysis. This lack of data poses challenges in building prognostic models. On the other hand, it is difficult to determine a maintenance strategy at a system level, considering limited and inaccurate prognostic information. Integrating multiple data sources, such as predicted RUL, economic dependencies, and time-to-failure knowledge, into system-level maintenance optimization remains a challenge. These limitations form the main focus of this thesis.

To bridge these gaps, the overarching objectives of this thesis are twofold, focusing on prognostics at the component level and making maintenance decisions at the system level. In the first three topics, deep learning methods are developed for prognostics on machines under different data availability scenarios. Specifically, when there is limited failure data but a larger amount of suspension data, a semi-supervised learning method is proposed to predict RUL by utilizing both types of data. Additionally, a novel method based on transfer learning is proposed for cross-domain prognostics, allowing for RUL predictions of machines operating in new conditions with only a small amount of suspension history data collected. Finally, a RUL prediction method based on state-space modeling and reinforcement learning is presented for situations where no data can be collected from machines operating in new environments. This method enables us to predict the RUL for machines operating beyond historical records.

The last two topics explore strategies for maintaining mechanical systems using prognostic information. The main focus is on addressing the challenge of having limited monitoring capability for the system and inaccurate predictions of RUL during maintenance schedules. A maintenance optimization approach is proposed that considers not all components can be monitored. This method helps in identifying predictive thresholds and optimizing planned inspection together for the entire system. In addition, a novel maintenance basis is introduced to determine preventive maintenance actions when prognostics are not sufficiently accurate. By incorporating error modeling into the optimization model, insights into the impact of prognostic errors on maintenance costs are obtained.

The research conducted in the thesis provides novel techniques for prognostics and maintenance decision-making using condition monitoring data. The developed methods will contribute to reducing the expenses associated with the operation and maintenance of mechanical systems.

Preface

The material presented in this thesis is based on the original work by Rui He. As detailed in the following, material from all chapters of this thesis has been published in five refereed journal articles under the supervision of Dr. Zhigang Tian and Dr. Ming J. Zuo in review and editing, project administration, and providing resources. The generated journal articles with Rui He as the first author since Rui He proposed all these research topics and did the primary research work, Dr. Zhigang Tian as the corresponding author, and Dr. Ming J. Zuo as a co-author. Mr. Yifei Wang, Ms. Yinuo Chen, and Mr. Ziwei Guo are coauthors for some publications.

The first three works presented in Chapter 2, Chapter 3, and Chapter 4, respectively, have been published as journal papers on Mechanical Systems and Signal Processing, which are as follows.

- He R, Tian Z, Zuo M J. A semi-supervised GAN method for RUL prediction using failure and suspension histories. Mechanical Systems and Signal Processing, 2022; 168: 108657.
- He R, Tian Z, Zuo M J. A transferable neural network method for remaining useful life prediction. Mechanical Systems and Signal Processing, 2023; 183: 109608.
- He R, Tian Z, Zuo M J. Machine prognostics under varying operating conditions based on state-space and neural network modeling. Mechanical Systems and Signal Processing, 2023; 182: 109608.

The work presented in Chapter 5 has been published as a journal paper on Reliability Engineering & System Safety, which is as follows. The contributions of coauthors Mr. Yifei Wang and Mr. Ziwei Guo were on the manuscript revision.

- He R, Tian Z, Wang Y, Zuo M J, Guo Z. Condition-based maintenance optimization for multi-component systems considering prognostic information and degraded working efficiency. Reliability Engineering & System Safety, 2023; 234: 109167.

The work presented in Chapter 6 has been submitted as a journal paper on Applied Energy, which is currently under review. The contributions of the co-author, Mr. Yifei Wang and Ms. Yinuo Chen, were on the manuscript revision.

- He R, Tian Z, Wang Y, Chen Y, Zuo M J. Condition-based maintenance for wind farms with imperfect prognostic information. Applied Energy, 2023 (Under review).

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my supervisors, Dr. Zhigang Tian and Dr. Ming J. Zuo, for their invaluable support, guidance, and motivation during my PhD study. Their vast knowledge and profound understanding have been a great source of inspiration for me throughout my PhD journey. I would also like to thank my supervisory and examining committee members, Dr. Di Niu, Dr. Rafiq Ahmad, and Dr. Nooshin Salari, for their valuable comments and suggestions on my thesis. I am grateful to Dr. Zengtao Chen for being my examination committee chair and dedicating his precious time to evaluating my thesis.

I would like to thank all the members of the Reliable Research Laboratory, including Dr. Xingkai Yang, Mr. Yifei Wang, Mr. Ziwei Guo, Ms. Yinuo Chen, and everyone else. I am grateful for their help, advice, and encouragement. Working and studying with them has been an exceptional journey for me. In particular, I am very grateful for the opportunity to have met Yifei. His assistance has greatly benefited both my personal life and academic endeavors. I extend my best wishes for his successful graduation and a lifetime of happiness with his loved ones.

Moreover, I consider myself very blessed to have met Dr. Shike Li on campus. I am very grateful for the support and care she has given me in my life. I sincerely wish her all the best in her life and career.

Finally, I am extremely appreciative to my family for their understanding, support, and endless love. No matter what happened, they always stood firmly behind me. Most importantly, I wholeheartedly wish my mother the best of luck and good health forever.

This is only the start of my research, not the end.

Table of Contents

List of Tables

List of Figures

List of Acronyms

AM	Age-based Maintenance
CBM	Condition-based Maintenance
CI	Constant Interval Maintenance
CM	Corrective Maintenance
CNN	Convolutional Neural Network
COSMO	Consensus Self-organizing Model
CV	Credibility Value
DANN	Domain Adversarial Neural Network
FOT	Fault Occurrence Time
FT	Failure Time
GA	Genetic Algorithm
GAN	Generative Adversarial Network
HI	Health Indicator
LSTM	Long Short-Term Memory
MAPE	Mean Absolute Percentage Error
MCS	Monte Carlo Simulation
MDP	Markov Decision Processes
MLE	Maximum Likelihood Estimation
MLP	Multilayer Perceptron
MMD	Maximum Mean Discrepancy
MSE	Mean Squared Error
NDT	Non-destructive Testing
NN	Neural Network
PCBM	Prognostics-induced Condition-based Maintenance
PDF	Probability Density Function
RKHS	Reproducing Kernel Hilbert Space
RMSE	Root Mean Squared Error
RUL	Remaining Useful Life
SD	Standard Deviation

SRL	State-space Modeling with Reinforcement Learning
SRRL	State-space Modeling with Robust Reinforcement Learning
SRGAN	Semi-supervised Regression Generative Adversarial Network
SUKF	State-space Modeling with Unscented Kalman Filter
TBM	Time-based Maintenance
TCA	Transfer Component Analysis
TMP	Transfer Multi-layer Perceptron
t-SNE	t-distributed Stochastic Neighbor Embedding
UKF	Unscented Kalman Filter
VI	Varying Interval Maintenance

List of Symbols

Chapter 1	
$x(t)$	The input vector of LSTM at time *t*.
$h(t)$	The output vector of LSTM at time *t*.
$i(t)$, $f(t)$, $o(t)$	The activation vectors of input gate, forget gate, and output gate.
$c(t)$	The cell activation vector.
σ	The Sigmoid activation function.
$\odot$	The element-by-element product.
W, b	The weight and bias of LSTM related to the different activation vectors.
$D(x;\ \theta_d)$	The discriminator parameterized by θ_d with input x sampled from distribution $p_{data}(x)$.
$G(z;\ \theta_g)$	The generator parameterized by θ_g with output p_g and input z sampled from noise distribution $p_z(z)$.
$V(G, D)$	The single objective function of GAN.
θ_d^*	The optimal θ_d for the discriminator.
θ_g^*	The optimal θ_g for the generator.
δ	The temporal difference error.
r	The reward.
γ	The discounting factor.
s	The state.
$Q(\cdot)$	The current value function implemented by the critic.

Chapter 2	
D_l	The labeled dataset that contains *m* data points, denoted by $\{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$, where *x* is the measurements and *y* is the corresponding RUL.
D_u	The unlabeled dataset that contains *n* data points, denoted by $\{x_{m+1}, x_{m+2}, \ldots, x_{m+n}\}$.
L_D	The loss of the discriminator.
$L_{supervised}$	The loss function for the supervised training task.
$L_{unlabeled}$	The loss function for estimating the continuous label of unlabeled data.
$L_{generated}$	The loss function for judging the input data as real data or a generated one.

L_G	The loss of the generator.
D_f	The dataset of failure histories, and each history is denoted by $D_{f,i} = \{(x_{f,i,1}, t_{f,i,1}), (x_{f,i,2}, t_{f,i,2}), ..., (x_{f,i,lfi}, t_{f,i,lfi})\}$, where $x_{f,i,j} \in R^{1\times d}$ is the measurements, $t_{f,i,j}$ is the inspection time, and $t_{f,i,lfi}$, also indicated by $T_{f,i}$, is the failure time.
D_s	The dataset of suspension histories, and each history is denoted by $D_{s,i} = \{(x_{s,i,1}, t_{s,i,1}), (x_{s,i,2}, t_{s,i,2}), ..., (x_{s,i,lsi}, t_{s,i,lsi})\}$, where $x_{s,i,j} \in R^{1\times d}$ is the measurements, $t_{s,i,j}$ is the inspection time, and $t_{s,i,lsi}$, also indicated by $T_{s,i}$, is the suspension time.
L_{ft}	The loss function for minimizing the difference of the predicted failure time among all inspection points in one suspension history.
L_{st}	The loss function for ensuring the predicted failure time of suspension histories exceed their suspension time.
$L_{matching}$	The loss function for feature matching.
$r_{i,j}$	The RUL percentage of the j-th inspection point in the i-th history data.
$\Omega_{i,j}$	The k closest neighbors in failure histories D_l for the j-th inspection point in the i-th suspension history, in which one data $(x_{k,i,j}, r_{k,i,j})$ is selected randomly during model training.
$p_{data}(x_f, r)$	The joint probability distribution of the condition monitoring measurements and the corresponding RUL percentage in D_l.
$p_{unlabeled}$	The probability distribution of the monitoring measurements x_s in suspended data D_u.
$p(\Omega\|x_s)$	The conditional probability distribution of the neighbor set given a specific x_s.
λ	The weight of the tasks in the loss function.
$\mu_{i,j}$	The mean of $r_{i,j}$, the mean and variance of which is $\gamma_{i,j}$ and $\sigma_{i,j}^2 v_{i,j}^{-1}$, respectively.
$\sigma_{i,j}^2$	The variance of $r_{i,j}$, which follows a Gamma distribution with shape and scale parameters $\alpha_{i,j}$ and $\beta_{i,j}$.
n_t	The number of test data.
Chapter 3	
$D^{(S)}$	The m failure histories in the source domain, and each history is denoted by $D_i^{(S)} = \{(x_{i,1}^{(S)}, t_{i,1}^{(S)}), (x_{i,2}^{(S)}, t_{i,2}^{(S)}), ..., (x_{i,lfi}^{(S)}, t_{i,lfi}^{(S)})\}$, where $(x_{i,j}^{(S)}, t_{i,j}^{(S)})$ is the j-th inspection point of the i-th failure history, and $x_{i,j}^{(S)} \in R^{1\times d}$ is condition monitoring measurements and $t_{i,j}^{(S)}$ is the inspection time.
$D^{(T)}$	The n suspension histories in the target domain, and each history is denoted by $D_i^{(T)} = \{(x_{i,1}^{(T)}, t_{i,1}^{(T)}), (x_{i,2}^{(T)}, t_{i,2}^{(T)}), ..., (x_{i,lfi}^{(T)}, t_{i,lsi}^{(T)})\}$, where $(x_{i,j}^{(T)}, t_{i,j}^{(T)})$ is the j-th

	inspection point of the *i*-th suspension history, and $x_{i,j}^{(T)}$ and $t_{i,j}^{(T)}$ are condition monitoring measurements and inspection time.
$y_{i,j}^{(S)}$	The RUL percentage of the *j*-th inspection point in the *i*-th failure history.
$t_{i,fo}^{(S)}$	The fault occurrence time of the *i*-th failure history.
$P(x)$	The marginal probability of *x*.
$Q(y\|x)$	The conditional probability of *y* given *x*.
$\phi(x)$	The domain invariant features.
τ	The number of extracted inspection points.
$\theta_{i,j}^{(S)}$	The COSMO feature of source-domain history.
$\theta_{i,j}^{(T)}$	The COSMO feature of target-domain history.
$\theta_{i,j,p}$	The *p*-th feature of the COSMO feature, where $p = 1,\ldots, d$.
$x_{H,q,p}$	The *p*-th feature of the *q*-th sample in the reference group *H*.
Th_{fot}	The threshold for identifying fault occurrence time.
Th_{ft}	The threshold for identifying failure time.
f	The NN-based regression model.
$L_{supervised}$	The loss function for the supervised training.
$L_{transfer}$	The loss function for domain adaptation.
$L_{manifold}$	The manifold regularization loss.
λ_t, λ_m	The weighting parameters.
λ_P, λ_Q	The weighting parameters.
$p_{data}(x^{(S)}, y^{(S)})$	The joint probability distribution of measurements and RUL percentage in $D^{(S)}$.
$p_{data}(x^{(T)}, y^{(T)})$	The joint probability distribution of measurements and RUL percentage in $D^{(T)}$.
$d(\cdot)$	The distance between the distributions in two domains.
$D_c^{(S)}, D_c^{(T)}$	The source and target domain data with the *c*-th class label, respectively.
n_c, m_c	The number of samples in $D_c^{(S)}$ and $D_c^{(T)}$ respectively.
L_i	The normalized graph Laplacian matrix for the *i*-th suspension history in the target domain.
I_i	The identity matrix with a dimension of $l_{s,i} \times l_{s,i}$.

D_i	A diagonal matrix for L_i given by $D_{i,j_1,j_1}=\sum_{j_2=1}^{l_{s,i}} W_{i,j_1,j_2}$, where W_i is the data adjacency graph.
$\mathcal{N}(x_{i,j}^{(T)})$	The set composed of the nearest neighbors of $x_{i,j}^{(T)}$.
d_p, d_q	The distance of the marginal probability distribution and the conditional probability distribution, respectively.
W_i	The data adjacency graph calculated based on *HI*.
$\hat{y}_{nt}$	The output of the *nt*-th built transfer learning network, where *nt* = 1, 2,…, *N*.
$\mu_{mc,i}$, $\sigma_{mc,i}^2$	The *i*-th sampled mean and variance.
$\bar{\hat{y}}_{nt}$	The mean of the predicted RUL values $\hat{y}_{nt}$.
l_{mc}	The length of the generated Markov chain.
Chapter 4	
D^S	The *m* failure histories in the source domain, and each history denoted by $D_i^{(S)}$ is $\{(x_{i,1}^S, w_{i,1}^S, r_{i,1}^S),(x_{i,2}^S, w_{i,2}^S, r_{i,2}^S),...,(x_{i,lsi}^S, w_{i,lsi,}^S, r_{i,lsi}^S)\}$, where $(x_{i,t}^S, w_{i,t}^S, r_{i,t}^S)$ is the *t*-th inspection point of the *i*-th failure history, and $x_{i,t}^S$ is condition monitoring measurements, $w_{i,t}^S$ indicates operating parameters, and $r_{i,t}^S$ is the corresponding RUL label.
D^T	The *n* target-domain histories, where the *i*-th target-domain history, D_i^T, denoted by $\{(x_{i,1}^T, w_{i,1}^T, r_{i,1}^T),(x_{i,2}^T, w_{i,2}^T, r_{i,2}^T),...,(x_{i,lti}^T, w_{i,lti,}^T, r_{i,lti}^T)\}$, and $(x_{i,t}^T, w_{i,t}^T, r_{i,t}^T)$ is the *t*-th inspection point of the *i*-th target-domain.
$f(\cdot)$, $g(\cdot)$	Nonlinear functions.
$\varepsilon_{i,t}$	The stochastic noise at time step *t* for the *i*-th history.
$d_{i,t}$	The disturbance at time *t* caused by model errors and measurement noises, which is sampled from a Normal distribution with mean $\mu_{i,t}$ and variance $\sum_{i,t}=\sigma_d^2 \mathrm{I}_d$ during model training, where I_d is the identity matrix with the dimension same to $\mu_{i,t}$, and σ_d is the preset standard deviation.
$\tilde{x}_{i,t}$	The estimated measurement error at time step *t* for the *i*-th history calculated by $x_{i,t}$ minus $\hat{x}_{i,t}$, where $\hat{x}_{i,t}$ is the estimated measurements.
$\tilde{r}_{i,t}$	The estimated RUL prediction error at time step *t* for the *i*-th history calculated by $r_{i,t}$ minus $\hat{r}_{i,t}$, where $\hat{r}_{i,t}$ is the predicted RUL.
$K_{i,t}$	The estimator gain at time *t* for the *i*-th history.

$\Delta\hat{r}_{i,t}$	The RUL increment at time *t* for the *i*-th history.
$\pi(\cdot)$	The neural network used to approximate $\Delta\hat{r}_{i,t}$.
$P(\cdot)$	The state transition function.
$d(\cdot)$	The disturber neural network used to approximate the disturbance.
$Q^{\pi}(\cdot)$	The Q-function of $\pi(\cdot)$.
$C_{i,j}$	The cost function at time step *j* in the *i*-th history data.
γ_r	The discounting factor for $\pi(\cdot)$.
$\|\cdot\|_2$	The second-order norm.
λ_r	The weighting factor for $\pi(\cdot)$.
$Q^{d}(\cdot)$	The Q-function of $d(\cdot)$.
$\mathcal{M}_r$, $\mathcal{M}_d$	The replay memory.
L_{actor}	The loss function of actor network.
L_{critic}	The loss function of critic network.
η	The $H\infty$ norm that quantifies the effect of disturbances on prognostic performance.

Chapter 5	
$c_{i,j}$	The *i*-th component in the *j*-th sub-system.
n_j	The number of components in the *j*-th sub-system.
N	The number of sub-systems.
$r_{i,j}$|	The real RUL percentage of component $c_{i,j}$.
$\hat{r}_{i,j}$	The predicted RUL percentage of component $c_{i,j}$.
T	Planned operating period.
T_b	The baseline working time with the maximum efficiency.
γ	The reward rate at maximum efficiency.
C_{to}	Total maintenance cost.
$x_{i,j}$	The condition measurements of component $c_{i,j}$.
$RUL(\cdot)$	The nonlinear prognostic model.
$Err_{i,j}(1-r_{i,j})$	The prognostic error of component $c_{i,j}$ when its RUL percentage is $r_{i,j}$.
θ	The drift parameter.
σ	The diffusion parameter.
Y	Standard Normal distribution.

t_m	Planned inspection interval.
p_{pm}	Preventive RUL percentage threshold.
p_{om}	Opportunistic RUL percentage threshold.
n_m	The number of planned inspections.
$C_{r,i}$	Replacement cost for the *i*-th component.
C_{fs}	Setup cost when CM is prepared.
C_{ps}	Setup cost when preventive maintenance are prepared.
$C_{m,k}$	The *k*-th maintenance cost during the planned period.
L	The loss of system efficiency.
$L_{i,j}$	The system efficiency loss due to the degradation of component $c_{i,j}$.
e	The system working efficiency.
M	The number of historical data.
$n_{L,t}$	The number of system efficiency loss data *L* at time *t*.
Ω	Efficiency model parameter sets.
$w_{i,j}$	The weight of the impact of degradation of component $c_{i,j}$ on the system efficiency.
n_T	The number of the discretization intervals.
$T_{F,i,j}$	The failure time of component $c_{i,j}$.
$l_{i,j}$	The lifetime of component $c_{i,j}$.
b	Imperfect maintenance factor.

Chapter 6

$c_{i,j}$	The *i*-th component in the *j*-th wind turbine.
M	Number of components in a wind turbine.
N	Number of wind turbines.
$\hat{t}_f$	the predicted failure time.
$t_{f,u}$	the upper limit of the estimated range of failure time.
$t_{f,l}$	the lower limit of the estimated range of failure time.
N_P	Parent node.
N_C	Child node.
N_V	Virtual evidence node.
S_q	The q-th state in the child node, q = 1, 2, …, Q.

$f_u(x)$	Probability distribution function of normal distribution with mean one and standard deviation σ_u.
$f_d(x)$	Probability distribution function of normal distribution with mean zero and standard deviation σ_d.
Po_u	Posterior upward probability.
$T_{F,i,j}$	Real failure time of component $c_{i,j}$.
$r_{i,j}$	Real RUL percentage of component $c_{i,j}$.
$\hat{r}_{i,j}$	Predicted RUL percentage of component $c_{i,j}$.
$\hat{r}_{Po,i,j}$	Estimated posterior RUL percentage of component $c_{i,j}$.
$\hat{t}_{f,Po,i,j}$	Updated prediction of failure time for component $c_{i,j}$.
$t_{i,j}$	Age of turbine component $c_{i,j}$.
$D(t)$	Degradation of turbine components at time t.
$\varphi(t; \mathrm{K}, \Theta)$	Base degradation function given deterministic set K and stochastic set Θ.
θ, β	Two parameters defined within Θ.
$\varepsilon(t; \sigma)$	Error term with volatility σ.
Λ	Pre-defined degradation threshold.
$\nu(\Theta)$	Posterior distribution of the stochastic parameter set Θ.
$RMSE_i$	Prognostic error of the i-th monitored turbine component.
$D_{e,i}$	Database of generated prognostics histories for the i-th monitored turbine component.
t_S	Simulation time.
T_{Max}	Maximum simulation time.
Δt	Simulation time interval.
t_L	Maintenance lead time.
n_f	Number of accidental failures.
τ_f	Acceptable number of failures.
n_I	Index for planned inspection, $n_I = 1, 2, \ldots, N_I$.
C_E	Maintenance cost per unit of time.
C_T	Total maintenance cost.
$\Delta C_{T,F}$	Total cost for failure replacement.
$\Delta C_{T,I}$	Total cost for inspection.
$\Delta C_{T,P}$	Total cost for preventive replacement.

$\Delta C_{T,O}$	Total cost for opportunistic replacement.
$\Delta C_{T,ts}$	Total maintenance cost at t_S.
$C_{P,i}$	Preventive replacement cost for the i-th turbine component.
$C_{R,i}$	Part cost for the i-th turbine component.
$C_{F,i}$|	Failure replacement cost for the i-th turbine component.
$C_{I,i,j}$	Planned inspection cost for the turbine component $c_{i,j}$.
$C_{P,i}^{S}$	Setup cost in the preventive replacement cost $C_{P,i}$.
$C_{F,i}^{S}$	Setup cost in the failure replacement cost $C_{F,i}$.
$C_{I,i}^{S}$	Setup cost in the planned inspection cost $C_{I,i,j}$.
C_A	Cost to access a turbine.
C_{Pre}	Preparation cost.
$I_{R,i,j}$	Binary variable indicating whether $c_{i,j}$ needs to be replaced during planned inspection.
$I_{F,i,j}$	Binary variable indicating whether $c_{i,j}$ fails.
$I_{F,i,j}^{Insp}$	Binary variable indicating whether $c_{i,j}$ needs to be inspected during failure replacements.
$I_{P,i,j}$	Binary variable indicating whether $c_{i,j}$ needs to be preventively replaced.
$I_{P,i,j}^{Insp}$	Binary variable indicating whether $c_{i,j}$ needs to be inspected during preventive replacements.
$I_{O,i,j}$	Binary variable indicating whether $c_{i,j}$ needs to be opportunistically replaced.
$I_{O,i,j}^{Insp}$	Binary variable indicating whether $c_{i,j}$ needs to be inspected during opportunistic replacements.
$I_{A,j}$	Binary variable indicating whether replacement actions will be performed on the j-th turbine.
I_{Pre}	Binary variable indicating whether any wind turbine needs to be maintained.
δ	Measure of inspection biases.

Chapter 1: Introduction

This chapter consists of three sections. In Section 1.1, the background of this thesis is introduced. Section 1.2 presents a review of the literature, from component-level prognostic techniques to system-level maintenance optimization. The thesis objective, research topics and their contributions, and thesis organization are described in Section 1.3.

1.1 Background

Mechanical systems typically contain multiple sub-systems, in which the principal sub-systems are specific machines designed expressly to perform specific tasks, such as the transference and transformation of motion, force, or energy, and other sub-systems are auxiliaries integrated to support the actions and work of the machine [1]. With advances in manufacturing techniques, mechanical systems have become more complicated in design, function, and management. Often, contemporary mechanical systems are mechatronic in nature, and their mechanical parts are usually much less reliable, creating comparatively the greatest risk in system operations and economic loss [2]. As examples of such mechanical systems, we may take (a) a wind turbine for energy generation, (b) a truck with positive displacement pumps for petroleum extraction, (c) a turbofan engine supplying power for airplane moving, or (d) a machine tool for metal processing in the case of mechanical engineering shown in Fig. 1.1.

The degradation of mechanical systems, or the aging or evolution of potential failures, may lead to a decrease in their working efficiency and an increase in unanticipated accidents. As reported, one out of every 100 road accidents is associated with mechanical failure of the braking system or electronic steering [3]. The rate of generation decline of wind farms is about 1%/year in the US [4] and more than 1.5%/year in the UK [5] due to equipment aging. A proper maintenance strategy can play a key role in improving system safety, efficiency, and economics by slowing the rate of equipment degradation and preventing component failure through timely inspection, repair, and replacement. Due to increasing automation and reliance on expensive equipment, both the proportion of employees performing maintenance work and the cost of maintenance are growing [6]. In current process and chemical industries, up to 30% of the workforce is engaged in

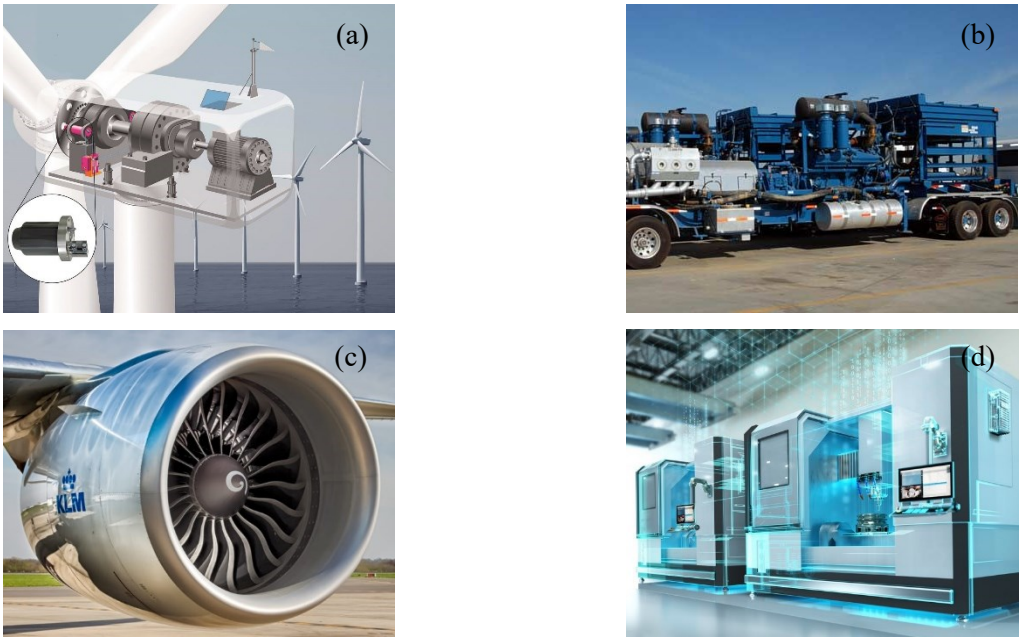


Fig. 1.1: Examples of the scope of research subjects

maintenance activities [7]. Maintenance costs typically account for 15–70% of the total expenditure of end products, making them one of the largest cost components in engineering assets [8]. As the modern industry moves toward data-driven "Industry 4.0", many companies have started to revise their maintenance policies to cope with fierce competition [9]. Theoretically, an optimal maintenance strategy can enhance economic benefits by up to 20% [10]. As an illustration, however, domestic plants in the US spent more than $600 billion over 20 years to maintain their critical systems, but more than one-third of this expenditure is wasted on ineffective maintenance [11]. Increasingly complex mechanical systems improve the potential for maintenance benefits and also create additional challenges for maintenance scheduling.

Many firms are still collecting equipment failure time data on an ongoing basis to guide system maintenance decisions. The failure time data is the termination time when the equipment stops functioning due to failure. Oftentimes, failure time gathers statistical information about the amount of time that a machine lasts before failure, and these statistics collected from a large sample of machines can be used to estimate the time-to-failure distribution and schedule maintenance for individual equipment. This traditional failure-time-based maintenance decision-making approach often determines maintenance actions periodically according to constant inspection intervals and is solely dependent on historical statistical information and the passage of time or measures of

usage of machines [12]. For example, an elevator may be inspected every six months. The manufacturer may recommend that the automobile's timing belt be replaced after five years or 60,000 miles. Traditional maintenance policies are easy to implement and are applicable particularly to non-mechatronic systems, e.g., power distribution systems [13], as their health status is hard to measure indirectly. Nevertheless, a large number of unnecessary maintenance actions may still be performed periodically when the mechanical system is in reasonable condition, leading to an inevitable increase in maintenance costs [14].

With advances in monitoring techniques, modern mechanical systems have incorporated various sensors to gather different types of condition monitoring data. These sensors, such as those for vibration, oil analysis, temperature, and pressure, can effectively capture the degradation process of machinery. The installation of monitoring sensors can observe the health status and operational characteristics of mechanical components in real time, enabling the early detection of abnormalities. As an example, accelerometers are widely employed to measure the vibration of the gearbox system, as the degradation of gears is usually accompanied by an increase in vibration amplitude [15]. Additionally, other sensors like thermistors, acoustic emission sensors, and current sensors are frequently utilized [16].

Condition-based maintenance (CBM) is a maintenance policy that utilizes condition monitoring data to make maintenance decisions. In 2006, Jardine et al. [17] outlined three key steps in a CBM program: data acquisition, data processing, and maintenance decision-making. In the second step, diagnostics and prognostics are crucial in improving the benefits of CBM. Diagnostics involves automatically communicating the location and type of fault to asset managers based on machinery condition monitoring data after a fault has occurred. In contrast, prognostics deals with fault prediction, aiming at predicting the time the machine can work until the next fault, i.e., the remaining useful life (RUL). Some companies are now integrating diagnostics into their maintenance decision-making process. For instance, Baker Hughes performs early repairs or replacements on pumping equipment based on incipient fault detections to ensure uninterrupted oil extraction [18]. Similarly, Vestas deploys maintenance on wind turbines once the detected equipment wear exceeds the unacceptable threshold [19]. Compared to diagnostics, prognostics is an analysis technique done prior to the occurrence of an event. Especially benefiting from big data

and artificial intelligence techniques, prognostics makes maintenance scheduling more efficient and can achieve zero-downtime performance [17].

In the study mentioned in [20], accurate RUL predictions can reduce routine maintenance work by 40–70%. However, increased prognostic error may lead to a rapid progression of CBM benefits in the negative direction [14]. In modern mechanical systems, a vast array of sensors is strategically placed to capture the condition monitoring data of various components. This has led to the proposal of various models, ranging from physics-based to data-driven methods, for predicting RUL [16]. Due to the complexity of failure physics, data-driven prognostics has become more and more attractive. Typically, artificial intelligence techniques, such as machine learning or deep learning, are widely used to discover the degradation information from equipment condition monitoring data [21]. Additionally, the computational capabilities of machines have advanced, allowing for numerical simulations that closely mirror real-world maintenance actions. These advancements provide more opportunities to utilize prognostic techniques in supporting system maintenance through real-time data.

However, to effectively schedule maintenance plans using prognostic information, it is essential to address real-world problems at both the component and system levels. Specifically, at the component level, it is necessary to develop prognostic techniques that can handle situations where there is limited data available further. There are two types of data commonly used for building prognostic models: failure history and suspension history. Failure history refers to the condition monitoring data collected from the beginning to the end of the equipment life. However, if engineering assets are taken out of service before experiencing complete failures, the data collected during this period is known as “suspension history”. In practice, engineering assets are often replaced before failures, resulting in only a few completed failure histories being available for collection [22]. The lack of data quantity and the presence of various working conditions pose challenges in building prognostic models for different tasks. On the other hand, it is still difficult to guide the maintenance policy of the whole complex system at the system level, considering limited and inaccurate prognostic information. Most existing studies on prognostics-induced CBM consider that all sub-systems can be monitored in real-time and primarily focus on finding optimal thresholds for maintenance based solely on accurate RUL predictions. However, in certain systems like wind turbine drive systems, only specific components like bearings and gearboxes can be

continuously monitored, while others like rotor-blades can only be detected during inspections. Furthermore, the predictions made by many prognostic models often deviate significantly from reality, especially when historical data is insufficient. The effective integration of multiple data sources, such as predicted RUL, health status from inspections, economic dependencies, and time-to-failure knowledge, into system-level maintenance decision-making remains a challenge. These limitations are the main focus of this thesis.

1.2 Literature review

If enough failure history is available, a variety of models can be developed to perform prognostics and schedule maintenance accurately. However, failure histories are usually limited. This section presents an examination of the existing literature on using limited condition data for predicting RUL at the component level and optimizing maintenance at the system level. This section consists of three subsections. Section 1.2.1 and Section 1.2.2 discuss studies that have been conducted on building prognostic models with suspension histories and under different operating conditions, respectively. Section 1.2.3 reviews the policies and methods that have been utilized for making maintenance decisions with prognostic information.

1.2.1 Prognostics of machines using suspension history

The condition data history of equipment refers to the condition monitoring data collected from the beginning to the end of the equipment life and used to provide data support for the RUL predictions. Most prognostic methods require abundant failure histories, in which equipment will end up with a failure and eventually be replaced. However, in practice, engineering assets are generally taken out of service before complete failures, either during planned maintenance or when an incipient defect is detected. Therefore, only limited failure histories can be collected. When an equipment function is terminated before failure, the exact failure time is unknown because the equipment has not completely failed. As defined by [22], the monitoring history of such equipment can be called "suspension history", or "right-censored history".

In conventional reliability assessment, only the terminated time of monitoring histories, i.e., failure time or suspension time, can be used to estimate the lifetime distribution for RUL predictions. When time data are sufficient, the lifetime distribution can be estimated as a Weibull distribution

by maximum likelihood estimation (MLE) [23]. Assuming that the equipment with suspension histories fails when actual degradation reaches a specified critical level, a two-stage statistical method can also be used to estimate the parameters of a degradation model for RUL predictions [24]. In contrast, when the failure information is incomplete, Bayesian inference formulated in the Markov chain Monte Carlo can provide acceptable effects for lifetime distribution estimations [25]. Besides, the Bayesian method can still get reliable estimates when using truncated time data [26]. Nevertheless, it is hard for the conventional reliability assessment methods to assess a complex distribution for lifetimes, and since the monitoring data in history is neglected, the equipment dynamic response and damage propagation cannot be concerned.

Data-driven prognostic methods focus on using equipment histories to model a one-to-one relationship between condition monitoring data and the corresponding RUL. However, due to the unknown failure time, suspension histories without RUL labels cannot be used directly. To address this issue, some works attempt to construct a degradation model first to predict the degradation trend of suspension history. When the degradation reaches the preset threshold, the corresponding time point will be defined as the failure time of suspension history, and then the suspension history can be used for further regression process. Typically, dynamic time warping [27] and support vector machines [28] have been used to predict the future degradation trend for suspension histories. In addition, Lu et al. [29] provided an unsupervised method based on the self-organizing map to extract the degradation feature from suspension histories without any knowledge of the degradation model. Xiao et al. [30] proposed a similar method to predict the future trend of monitoring data online by setting a time window and using a neural network, but without pre-training the model in advance. Intuitively, the predicted failure time of suspension histories will strongly depend on a suitable threshold and a large amount of data. This means that the methods mentioned above can only provide satisfactory performance under the conditions that are fully documented in the training dataset.

As a subfield of data-driven methods, deep learning has the capability to extract the dynamic and hierarchical representation of the degradation process and thus can be a powerful solution for RUL predictions [21][31][32]. When the amount of labeled data is limited, the semi-supervised learning architecture can be used to make full use of other available data and basic deep learning models to improve the model performance [33]. The prognostic methods based on semi-supervised learning

focus on using the condition monitoring data without RUL labels to improve the model generalizations. The suspension history can be commonly regarded as the unlabeled data in semi-supervised RUL prediction methods. For instance, Hu et al. [34] constructed two regressors based on co-regression to predict the RUL of suspended data and train the regression model using the data with RUL labels simultaneously. Ellefsen et al. [35] first used all histories to train a restricted Boltzmann machine model as an encoder and used failure histories to build a long short-term memory (LSTM) as a decoder to achieve effective RUL prediction. Similarly, Yoon et al. [36] utilized a variational autoencoder as the encoder and recurrent neural networks as the decoder but further predicted the RUL of suspended data based on self-learning and retrained the model by using all histories with RUL labels. Even though the accuracy can be enhanced in RUL prediction, the drawback of these semi-supervised learning approaches lies in the fact that the predicted RUL of all inspection points in one suspension history is not dependent, resulting in incorrect or even physically unreasonable RUL estimates. Tian et al. [22] proposed a practical model training approach similar to self-learning, but the difference lies in adding a constraint to the predicted failure time of suspension histories to ensure that the failure time is always greater than the suspension time to satisfy the physical rationality. You and Meng [37] presented a self-learning framework based on the similarity measure to estimate the failure time of suspension histories and build a regression model. As pointed out in [38], however, the predicted failure time of suspension histories depends only on failure histories and one regressor, leading to poor model generalization, especially when the amount of data is limited.

To summarize, conventional reliability methods rely only on failure or censored time data to estimate machinery failure time. Therefore, these methods are unable to provide accurate prognostics based on data monitoring. Semi-supervised learning methods, on the other hand, can utilize both failure and suspension histories as labeled and unlabeled data. However, these methods overlook valuable information in suspension histories, which makes it challenging to deliver reliable and convincing prognostics. To address this limitation, Chapter 2 of this thesis aims to develop a semi-supervised learning method that takes into account the information from suspension histories for improved prognostics.

1.2.2 RUL predictions of machines under varying operating conditions

Most prognostic methods require a large number of failure histories that can cover all possible operating conditions. There are different types of models available for predicting the RUL of machines. These models range from physics-based approaches to data-driven methods. If a physical degradation model can generalize well to all possible operating conditions, this model can be tuned based on different operating conditions using model updating methods like Bayesian inference [39] and particle filter [40][41]. However, building and implementing authentic physics models can be challenging due to the stochastic and complex nature of machine degradation processes. Alternatively, the physical system can be represented mathematically using the state-space model. This model consists of input, output, and state variables that are related by time-dependent difference equations. When the physical degradation process is unknown, the machine degradation characteristics, such as crack or wear size, can be considered as a state variable. The model parameters can then be determined based on historical condition data using data-driven methods. The state-space of a degradation process can be modeled using statistical and stochastic processes or model-free functions, such as the Gaussian model [42], Wiener process [43][44], and neural networks [45][46].

Taking into account the various conditions under which equipment operates, Kundu et al. [47] proposed a Weibull accelerated failure time regression model. This model considers both operating parameters and condition monitoring signals when estimating parameters. Li et al. [48] applied a linear parameter-varying model to identify the degradation state-space that varies with operating parameters. The Wiener process was also developed to incorporate time-varying operating conditions in the state-space modeling [49][50]. The state-space model can provide a clear interpretation. However, state-space-based prognostic approaches define RUL as the time until the degradation level of machines reaches or exceeds a predetermined failure threshold. A state-space model often assumes that the degradation mechanism remains unchanged under different operating conditions, allowing for a subjective and constant failure threshold to be set for all possible cases. In addition, the construction of state-space models requires complete training data that covers all possible operating conditions. These limitations make existing state-space models less suitable for predicting RUL in equipment operating under new conditions.

A deep learning-based prognostic model is commonly constructed offline and used to predict RUL when new condition data is monitored. In deep learning-based prognostic models, the operating parameters are often included as one part of the inputs. The one-to-one relationship between the condition monitoring signal and the corresponding RUL can be established and learned using failure histories. Different deep learning methods, such as autoencoder [51], long short-term memory (LSTM) [52], and convolutional neural network (CNN) [53], can be used to model the nonlinear prognostic function. Advanced deep learning methods like conditional variational autoencoder [54], bidirectional LSTM [55], and temporal convolutional network [56] have also been developed to handle multiple operating conditions. If a prognostic model captures all possible working conditions, the model can be well adapted to the various situations of the machine. However, traditional deep learning methods assume that the measurements of training and testing samples follow a similar distribution, which may not hold true for equipment with a wide range of unrecorded working conditions.

Transfer learning methods have gained attention in recent years for prognostic and health management purposes [32]. These methods aim to reduce the differences in data distribution between training and testing samples by learning deep conjunct representations from data in two different domains. Many existing studies focus on scenarios where only unlabeled target-domain data can be collected. For instance, Fan et al. [57] proposed a transfer prognostic method called consensus self-organizing models (COSMO), which selects features between the source and target domains without considering the transferability of features. Sun et al. [58] developed a prognostic model based on a sparse autoencoder and transferred the model from the source to the target domain by minimizing the KL divergence between the two domains. Similarly, da Costa et al. [59] applied the concept of domain adversarial neural network (DANN) to train a cross-domain prognostic model that minimizes the divergence between the source and target data. DANN was also adapted as an online prognostic model that updates model parameters using online data [60].

Furthermore, Mao et al. [61] presented a framework for transfer prognostics that utilizes a denoising autoencoder as an encoder to extract degradation features. They minimized the maximum mean discrepancy (MMD) between these features in two domains to achieve domain adaptation. Ding et al. [62] also employed MMD for domain adaptation but focused on integrating few-shot learning to address scarce data issues. Zhu et al. [63] combined both domain adversarial

loss and MMD into a domain adaptation method and extracted domain features using fully connected layers. Cheng et al. [64], Ding et al. [65], and Fu et al. [66] proposed MMD-based adversarial domain adaptation methods, but they extracted degradation features using the CNN, kernel regression, and LSTM models, respectively. In addition, Li et al. [67] proposed a generative adversarial network (GAN)-based method that identifies the initial fault occurrence time for each history data and performs domain adaptation in a single neural network model with different tasks. Zhang et al. [68] considered censored target-domain data in transfer prognostics through degradation direction alignment. Ideally, a prognostic model based on transfer learning can utilize deep learning architectures to adapt to different domains and predict the RUL of equipment operating under new conditions. Even if the operating conditions of the target machine differ greatly from the training data, transfer learning methods can still generalize the equipment degradation characteristics from training conditions to the target operating conditions. However, domain adaptation necessitates condition monitoring data for the target operating conditions of machines, even if these data are unlabeled.

Although some progress has been made, there are still limitations hindering the successful application of prognostic models in new environments. When we are able to gather some data from the target-domain machines, transfer-learning-based methods can be used to perform cross-domain prognostics. However, most of the relevant work simply aligns the degradation history of the two domains in a holistic manner. Therefore, these studies can only predict the RUL of machines under identical working conditions but fail to address the prognostic problem under multiple working conditions. Chapter 3 of this thesis focuses on exploring how target-domain data can be utilized for machine prognostics across different conditions. When we are unable to have access to target-domain data, existing methods are limited for prognostics of machines working in new environments. As a result, in Chapter 4, a robust prognostic model based on state-space modeling and reinforcement learning is proposed. This model can predict the RUL of machines operating under new conditions even without corresponding training data.

1.2.3 Maintenance decision-making for mechanical systems

Maintenance strategies commonly employed in the field of asset management include corrective maintenance (CM), time-based maintenance (TBM), and condition-based maintenance (CBM). CM entails performing maintenance actions solely when a component fails. On the other hand,

TBM and CBM are more advanced policies that incorporate fixed time intervals and condition estimates, respectively, to enhance maintenance efforts. Asset maintenance behaviors are often simulated in the framework of Monte Carlo simulation (MCS) [69][70] or Markov decision processes (MDP) [10][71]. Optimization algorithms such as genetic algorithm [72] are utilized to determine the most optimal maintenance decisions regarding the maintenance cost or profit.

Implementing TBM is a straightforward process that only requires recording the age or reliability information of in-service components. This information can be estimated by analyzing a large amount of time-to-failure data collected from components. When applying TBM to a complex mechanical system, it is important to consider the economic dependence of the system. This dependency is reflected in the fact that it is often cheaper to schedule maintenance for multiple components together during a single maintenance trip, rather than deploying maintenance separately [73]. This strategy is especially beneficial for certain systems where maintenance preparation is expensive. In earlier research, whenever a component triggers the maintenance action, the maintenance crew will take advantage of this opportunity to also perform preventive maintenance on other running components with relatively high age [10][74] or low reliability [75]-[77]. However, these studies only optimized inspection time intervals as the decision variable without considering the health conditions of machines. One major drawback of TBM is that it can lead to unnecessary maintenance actions being performed when the machine is still in good condition, resulting in inevitable increases in maintenance costs [14].

In contrast, CBM requires additional techniques to analyze the condition data collected from engineering assets but allows for more effective maintenance plans. As an illustration, benefiting from the inspection data, the inspection interval can be re-scheduled, e.g., in [78][79], according to the system status. Also, in [80][81], the degradation information can be used to investigate the optimal time for the next inspection. When implementing CBM in large mechanical systems, condition data is used to assess the health status of each component to identify suitable opportunities for group maintenance. In [82], when the fracture size of one blade in a wind turbine reached a certain threshold, other blades were also inspected. This integration of condition data and maintenance opportunity was suggested to be effective in reducing maintenance costs, although the findings are limited to a single component. Pérez et al. [83] expanded the opportunistic policy to include multiple components. This approach considered possible

dependencies within a wind turbine based on the health status determined by condition data. Ideally, CBM allows for maintenance actions to be taken just before the next equipment failure.

With the advancements in sensor and monitoring techniques, asset managers are now able to utilize continuous monitoring data to schedule maintenance actions in real time. When incorporating data monitoring, the goal of CBM policies is to determine one or two optimal thresholds that can be used to trigger preventive maintenance actions [17]. Some studies described the degradation behavior of machines by means of the proportional hazards model [84]-[86] and determined preventive maintenance actions based on the degradation values. Other works commonly used stochastic models, such as the Gamma process [87][88] or Wiener process [89]-[91], to represent machine degradation. In these studies, the parameters of degradation models were estimated using historical data. However, it is always a challenge to construct accurate degradation models and determine appropriate failure thresholds for all machines when there is insufficient relevance. In comparison to these methods, the prognostics-induced CBM policy uses predicted RUL to deploy maintenance actions [17]. This policy prioritizes preventive maintenance for components with a low RUL and allows for the simultaneous maintenance of multiple components with low RULs in large mechanical systems [74][92][93]. This integration of prognostics and opportunistic maintenance helps minimize downtime and preparation costs associated with individual maintenance actions. Some prognostic models can be created using a limited amount of failed or censored historical data [22]. This suggests that CBM policies that take into account prognostic information show promise. However, most existing studies assume that condition monitoring is available for all components in a system. Furthermore, while accurate RUL estimates have been shown to reduce routine maintenance by 40-70% [20], there is still a lack of understanding of how the accuracy of prognostic models affects the benefits of CBM, particularly for systems with multiple components.

Advances in monitoring techniques have led to the increased utilization of prognostic information to reduce maintenance costs in mechanical systems. However, if one follows the approach suggested in previous studies, which involves solely relying on the predicted RUL to make maintenance decisions, it would be unreliable in cases where the prognostic information is not accurate and comprehensive. In this thesis, Chapter 5 and Chapter 6 explore the integration of prognostics into maintenance decision-making. Chapter 5 specifically addresses the maintenance

challenges in a mechanical system where only certain components can be monitored. Building upon Chapter 5, Chapter 6 focuses on developing targeted maintenance optimization strategies for a more realistic system, i.e., wind farm, particularly when there are significant errors in the prognostic information.

1.3 Background knowledge

Prognostic models provide predictions of the RUL of components, which in turn help in making decisions about appropriate maintenance actions. Prognostic models are usually designed for individual components, and maintenance plans are created at the system level. In this section, three basic deep learning models are first introduced. These methods will be developed in this thesis for better RUL predictions. Lastly, the concept of MCS is illustrated, which plays a key role in the numerical simulation of maintenance cost.

1.3.1 Long short-term memory

LSTM networks are a variant of recurrent neural networks that have memory units incorporated into their structure. This allows LSTM networks to address the issue of gradient vanishing and overcome the limitations of traditional recurrent neural networks in learning long-term dependencies. This advancement in architecture makes LSTM networks well-suited for handling data with temporal correlation, which is beneficial for RUL predictions. The Fig 1.2 illustrates the architecture of LSTM, and the formulation of an LSTM unit is as follows:

$$i(t) = \sigma(W_{ix}x(t) + W_{ih}h(t-1) + b_i) \tag{1.1}$$

$$f(t) = \sigma(W_{fx}x(t) + W_{fh}h(t-1) + b_f) \tag{1.2}$$

$$o(t) = \sigma(W_{ox}x(t) + W_{oh}h(t-1) + b_o) \tag{1.3}$$

$$c(t) = f(t) \odot c(t-1) + i(t) \odot \tanh(W_{cx}x(t) + W_{ch}h(t-1) + b_c) \tag{1.4}$$

$$h(t) = o(t) \odot \tanh(c(t)) \tag{1.5}$$

where $x(t)$ and $h(t)$ denote the input and output vector at time t, $i(t)$, $f(t)$, $o(t)$ are the activation vectors of the input gate, forget gate, and output gate, respectively, $c(t)$ denotes the cell activation

vector, σ indicates the Sigmoid activation function, $\odot$ indicates the element-by-element product, and W and b represent the weight and bias related to the different activation vectors.

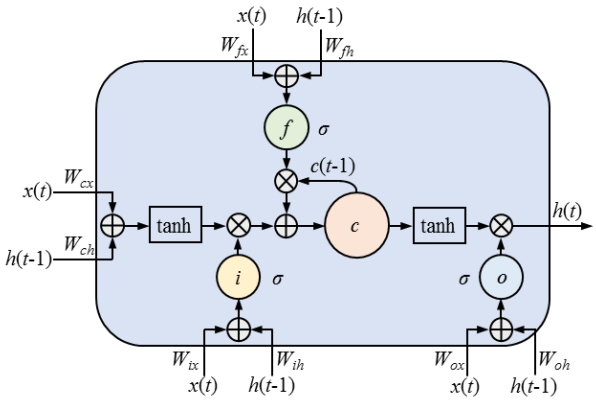


Fig. 1.2: Generic structure of an LSTM unit

1.3.2 Generative adversarial network

A GAN consists of two neural networks, i.e., a generator and a discriminator. The generator parameterized by θ_g tries to generate fake samples to fool the discriminator, while the discriminator parameterized by θ_d tries to distinguish between actual samples and artificially generated samples. Let x represent the training labeled data, and z denotes the random noise sampled from a normal distribution. In order to learn a generator distribution p_g over data x, the generator builds a mapping function from the prior noise distribution $p_z(z)$ to the training data space $p_{data}(x)$ as $G(z; \theta_g)$. Meanwhile, the discriminator $D(x; \theta_d)$ outputs a single scalar representing the probability that the input comes from the training data instead of p_g.

To minimize the divergence between p_g and $p_{data}(x)$, the log (1-$D(G(z))$) and log ($D(x)$) are respectively minimized and maximized by the simultaneous training of the generator and discriminator. Therefore, a basic GAN shown in Fig. 1.3 can be formulated as a two-player non-cooperative game with the mean value of cross-entropy losses by:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p\text{data}(x)} \log D(x) + \mathbb{E}_{z\sim pz(z)} \log(1 - D(G(z))) \tag{1.6}$$

where V (G, D) is the single objective function of GAN, log ($D(x)$) is the cross-entropy of the discriminator with real data x as input, and log (1-$D(G(z))$) is the cross-entropy of the discriminator with fake data $G(z)$ as input. Training GANs focuses on finding a Nash equilibrium in a two-player

non-cooperative game. The point (θ_d^*, θ_g^*) can be defined as the Nash equilibrium so that the discriminator and generator are both optimal simultaneously due to θ_d^* and θ_g^*.

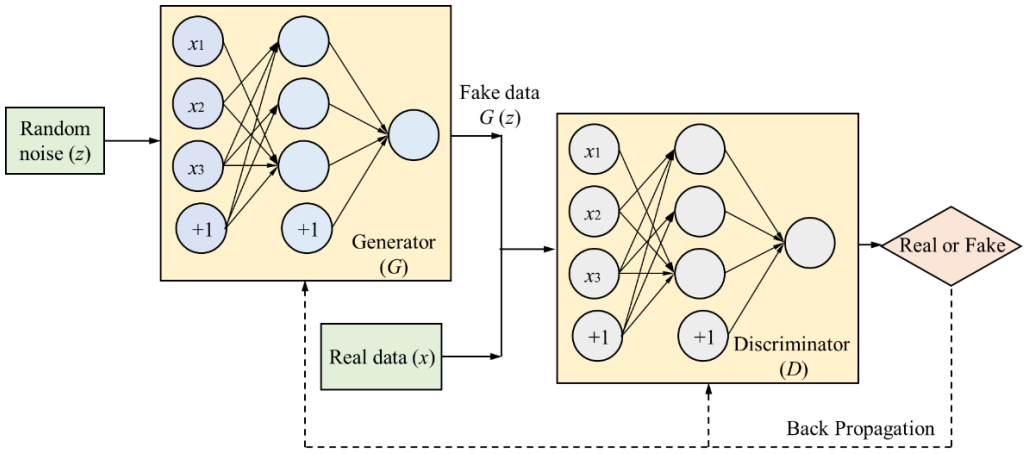


Fig. 1.3: Structure of basic GAN

1.3.3 Actor-Critic algorithm for reinforcement learning

Actor-critic methods are temporal difference learning methods that represent the policy function independent of the value function. In the Actor-critical approach, the policy is known as the actor, and it generates a range of possible actions based on the current state. On the other hand, the estimated value function is called the critic, and it evaluates the actions proposed by the actor using the given policy. The architecture of the actor-critic method is shown in Fig. 1.4.

The Critic is a function that estimates the value of a state. After each action selection, the Critic evaluates the new state to determine whether things have gone better or worse than expected. This can be formulated by:

$$\delta_t = r_{t+1} + \gamma Q(s_{t+1}) - Q(s_t) \tag{1.9}$$

where δ is the temporal difference error, r is the reward, γ is the discounting factor, and $Q(\cdot)$ is the current value function implemented by the critic. When an action is taken in a particular state s, the Critic uses this error to evaluate the selected action. A positive error suggests that the tendency to select that action should be reinforced in the future, while a negative error suggests that the tendency should be weakened. Both the Actor and Critic can be represented by neural networks and used for training the model.

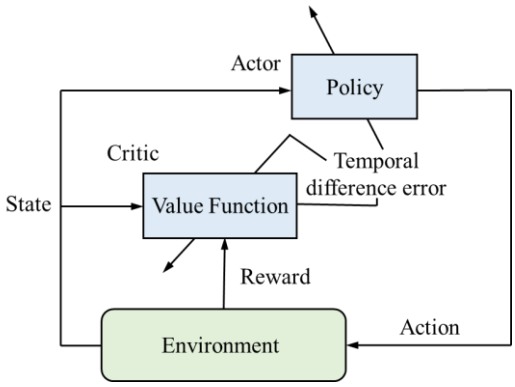


Fig. 1.4: Actor-critic architecture

1.3.4 Monte Carlo simulation

Monte Carlo simulation (MCS) is a type of computational method that relies on random sampling to obtain numerical results. It is used to solve problems that may have deterministic solutions by introducing randomness. MCS involves repeatedly sampling from a distribution to obtain a probabilistic understanding of the data. It has been extensively employed in fields such as reliability and risk assessment to propagate uncertainty. In this thesis, MCS is developed to represent the uncertainty of the time-to-failure distribution of components in optimizing system maintenance.

MCS methods typically involve two steps: 1) randomly generating particles from a probability distribution across the domain, and 2) performing a deterministic computation on these particles. When it comes to maintenance decision-making schedules, the goal is to optimize maintenance actions at the system level. Generally, certain key variables are identified to control maintenance decisions. By taking these variables into consideration, maintenance plans are determined for each component in various scenarios. To minimize maintenance costs and find optimal decisions, a maintenance optimization model can be formulated by minimizing the expected maintenance cost based on maintenance variables and the chosen policy. The numerical simulation method is a flexible approach for maintenance optimization as it takes into account strong dependencies and opportunities. In this scenario, MCS is used to estimate the maintenance cost by considering these maintenance variables. We generate random samples from the time-to-failure distribution of each component to simulate their lifespan from replacement to failure. This approach helps us evaluate the maintenance cost of the entire system through numerical simulations.

1.4 Thesis objective and outline

Deep learning has gained increasing attention in prognostics and health management because of its outstanding data processing capabilities that have the potential to enable better maintenance, repair, and overhaul activities [21]. Benefiting from the development of deep learning techniques, the overarching objective of this thesis is twofold. First, deep learning-based prognostic methods will be developed under different scenarios of data availability. Second, maintenance strategies will be designed for mechanical systems considering prognostic information. In this thesis, the following problems are addressed:

1) To incorporate the information of suspension histories into the prognostic model for better RUL predictions.
2) To predict the RUL of machines across different operating conditions based on limited target-domain suspension histories.
3) To predict the RUL of machines operating under new conditions without corresponding training data.
4) To find maintenance decisions at the system level, including optimized maintenance intervals and prognostic thresholds, under the assumption that the RUL of only certain components can be predicted.
5) To study the maintenance basis with partial and inaccurate prognostic information.

To achieve these objectives, five research topics have been completed in this thesis, which are presented as follows.

In the first research topic (Topic #1), a semi-supervised model called semi-supervised regression GAN (SRGAN) is developed to predict RUL by utilizing both failure and suspension histories. This model addresses the lack of failure history by extracting information from suspension histories, reducing the need for a large amount of failure history for training. The proposed method employs multi-task objective functions to capture valuable information from suspension histories rather than treating them as unlabeled data. This information includes: 1) ensuring similar predicted failure times for each inspection point in a suspension history, 2) taking into account the suspension time when the associated components were taken out of service, which should be earlier than their predicted failure time, and 3) considering the degradation reflected in the

condition monitoring data as the equipment deteriorates over time. The method will not directly estimate the times when suspensions fail, but instead compares the statistical information from similar failure and suspension histories to train the model. This means that the failure information of suspension histories is dependent not only on the failure histories but also on the generated data. Additionally, a method to evaluate the robustness of the prognostic model is proposed, which assesses the uncertainty caused by the scarce failure data. Therefore, the limitations of reliability assessments [23]-[26] and semi-supervised learning methods [34]-[37] in using suspension histories can be addressed. The materials of this research topic have been published in a journal paper [175].

In the second research topic (Topic #2), the RUL predictions of machines across different operating conditions are investigated. Machines often operate in various conditions that are not captured in the historical dataset. This poses a challenge as prognostic models trained on this data may not be able to accurately predict the RUL of machines in new environments. When the machine operates in a new environment, planned maintenance can be conducted to collect very few suspension histories. The goal of this study is to use these limited suspension histories to improve cross-domain prognostics. To address the limitation in [59]-[68] that only considers aligning two-domain histories in a holistic manner, the proposed approach simultaneously minimizes the differences in both the marginal and conditional probability distributions between different domains. To be specific, the first step is to create a health indicator (HI) using COSMO for the training data in both the source and target domains. This HI helps us estimate the fault occurrence time and the failure time to label the RUL for two-domain histories. In the second step, domain invariant features are learned in the transfer learning framework. A joint MMD method is proposed to learn generalized domain invariant features. This allows the model to adapt to different working conditions in terms of both marginal and conditional directions. Also, a heuristic method and a parallel framework are proposed to verify model parameters and uncertainties. The materials of this research topic have been published in a journal paper [176].

In the third research topic (Topic #3), the situation where no data in any form can be collected from a machine operating under new conditions is explored. The objective of this topic is to predict the RUL of machines operating under new conditions, even without having corresponding training data. Previous works, including model-based methods [39]-[50], supervised learning [51]-[56],

and transfer learning [59]-[68], require complete training data that covers all possible operating conditions, even though some data can be unlabeled. To overcome this limitation, this study proposes a new RUL prediction method that is based on state-space modeling and reinforcement learning, instead of directly constructing the relationship between the condition data and their RUL. On the basis of the state-space model, an interpretable prognostic model that combines Lyapunov constraint and reinforcement learning is proposed to predict the equipment RUL. Additionally, an adversarial training method based on the idea of $H\infty$ robustness is integrated to reduce the effect of state-space modeling errors. The key idea is to enhance the sensitivity of predicted RUL to unobserved degradation characteristics, e.g., cracks and wear, while reducing the sensitivity to operating parameters. This method improves its robustness to different conditions by reducing the causality between RUL and operating parameters and increasing the causality between RUL and unobserved degradation characteristics. Consequently, the proposed model is interpretable and capable of predicting RUL for equipment operating beyond historical records. The materials of this research topic have been published in a journal paper [177].

In the fourth research topic (Topic #4), a CBM optimization method is proposed for complex mechanical systems considering prognostic information and degraded working efficiency. Previous studies [84]-[93] have assumed that condition monitoring is available for all components in a system. However, in reality, only a few components can be monitored online in a system, while others need to be inspected manually during maintenance. Another limitation emerges when calculating profits in reported works. The impact of degradation on system profitability is rarely considered, whereas this impact is apparent in practice [4][5]. This research topic aims to overcome these limitations by finding a joint maintenance decision that maximizes the net revenue. This involves optimizing maintenance intervals and predictive thresholds, taking into consideration that prognostics is only available for certain components. Furthermore, the impact of system degradation on efficiency is investigated when evaluating the advantages of maintenance decisions. A model that considers the relationship between system efficiency and degradation is proposed, and the parameters of this model can be determined using historical data. We extend our model with prognostic error modeling to derive insights into the effects of the prognostic model on maintenance benefits. A case study involving the optimization of maintenance for wind turbine farms is presented to demonstrate and verify the proposed method. The materials on this topic have been documented in a journal paper [178].

In the fifth research topic (Topic #5), a maintenance optimization method is proposed for wind farms, taking into consideration imperfect prognostics. When the prognostic information is not accurate and comprehensive enough, it is unreliable to determine maintenance actions for wind turbines based only on predicted remaining useful life, as in existing studies [84]-[93]. To bridge this gap, a maintenance optimization strategy is developed for wind farms regarding condition monitoring with auxiliary inspection information. In this study, we determine predictive maintenance thresholds while optimizing the planned inspection intervals at the wind farm level. A new maintenance basis, denoted by the posterior RUL percentage, is proposed to calibrate prognostics with the help of rough estimates of the turbine component lifetime from inspections. When prognostics is confident, maintenance actions can be guided by the predicted lifetime alone. Conversely, standard predictive thresholds can also be optimized to complement time-based maintenance to prevent turbine component failures within two adjacent inspections, thus providing the potential for maintenance planning with partial and inaccurate prognostic information. The maintenance costs and policies are formulated into a large-scale optimization model to capture the economic dependencies between components within a turbine, as well as components across turbines in the farm. A numerical example is provided to demonstrate and verify the proposed method. Insights on the influence of prognostic error on maintenance costs are derived from comparative studies. The materials of this research topic have been submitted as a journal paper.

The novel contribution of this thesis is two-fold with respect to component-level prognostics and system-level maintenance, as summarized below:

1) **Prognostics:** Deep learning techniques are devised to address the issue of limited data availability in prognostics. Specifically, when there are only a few instances of failure data but a larger number of suspension data, an SRGAN method is proposed to predict the RUL by utilizing both types of data. In this method, a multi-task objective function is established to capture valuable information from suspension histories, allowing for more accurate estimates of RUL that align with physical expectations. Regarding the RUL prediction for machines operating in unrecorded conditions, two different scenarios are considered. In the first scenario, where very few suspension histories can be collected during planned maintenance for the machine working in a new environment, a novel method that combines COSMO and transfer learning is proposed for cross-domain prognostics. This method

allows the prognostic model to work for different conditions by adapting domain features from both marginal and conditional perspectives. As a result, it significantly improves RUL predictions using limited suspension histories. In the second scenario, where no data in any form can be collected from a machine operating under new conditions, a new RUL prediction method based on state-space modeling and reinforcement learning is presented. This model is trained using adversarial training based on the concept of $H\infty$ robustness. It is designed to be robust to operation parameters and sensitive to the RUL of equipment operating beyond historical records.

2) **Maintenance decision-making:** A comprehensive study is conducted to investigate how prognostics could be more practically integrated into system maintenance decision-making. The main contribution lies in dealing with the issue of having only partial monitoring capability for the mechanical system and the lack of accurate RUL predictions during maintenance scheduling. A maintenance optimization approach is proposed that takes into account the RUL prediction available only for certain components of the system. This method helps identify maintenance thresholds while also optimizing planned inspection intervals at the system level. During this process, a degradation-related system efficiency model is proposed to evaluate maintenance benefits more realistically. Furthermore, a new maintenance basis denoted by the posterior RUL percentage is presented to determine preventive maintenance actions. This maintenance basis combines auxiliary health estimates gathered during inspections with predictive analytics to calibrate RUL predictions, thereby enabling better decision-making to enhance maintenance benefits, especially in situations where prognostics is not perfect. Insights on the influence of prognostic errors on maintenance costs are derived by inducing error modeling into the optimization model.

The rest of this thesis is organized as follows. Chapter 2, Chapter 3, Chapter 4, Chapter 5, and Chapter 6 present the details of the four research topics, namely Topic #1, Topic #2, Topic #3, Topic #4, and Topic #5, respectively. Lastly, in Chapter 7, the research studies conducted in this thesis are summarized and future explorations are also discussed.

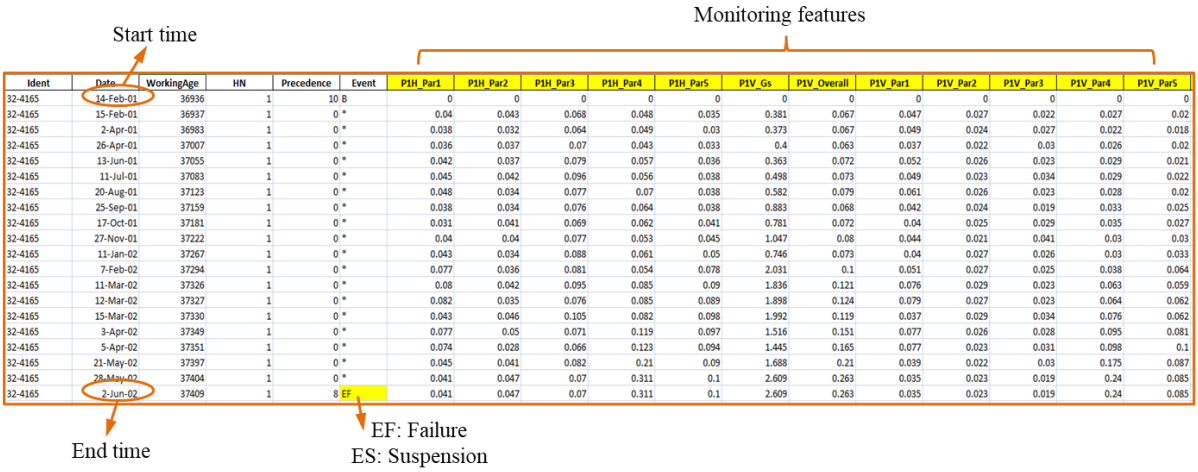
This thesis is written using the paper-based template, which meets the formatting requirements of the Faculty of Graduate Studies and Research at the University of Alberta.

Chapter 2: Semi-supervised GAN for machine prognostics using failure and suspension histories

Engineering assets are generally replaced by new ones before failure during planned maintenance, resulting in limited failure histories and often more than double the number of suspension histories. As suspension histories cannot be labeled with RUL values due to the unknown failure time, semi-supervised regression models are explored to incorporate unlabeled suspension histories in RUL predictions. This chapter is covered by the first research topic (Topic #1), as mentioned in Section 1.3. The organization of this chapter is as follows. In Section 2.1, the techniques for prognostics incorporating suspension histories are introduced. In Section 2.2, the concept of GAN and the basic principles of semi-supervised regression generative adversarial network (SRGAN) used for RUL predictions are presented. Section 2.3 describes the proposed RUL prediction method based on semi-supervised learning. Section 2.4 provides an in-depth analysis of the proposed model, especially for the model setting and uncertainty quantification. Section 2.5 uses two case studies to validate the proposed methods. Finally, conclusions are made in Section 2.6. The results of this chapter have been published as a journal paper [175].

2.1 Introduction

The condition data history refers to the collection of monitoring data that tracks the health status of the machine from the moment it is first used until it is retired or replaced after undergoing maintenance. There are two types of data commonly used for building prognostic models: failure history and suspension history. As illustrated in Fig. 2.1, a history involves three essential pieces of information: date, event, and monitoring features. If the event is marked as "*EF*", it means that the machine is replaced with failure. In this case, the end time is the failure time, and we can calculate the RUL for each line of monitoring features by subtracting the current time from the failure time. However, when an equipment function is terminated before failure, as marked by "*ES*", the exact failure time is unknown because the equipment has not completely failed. The monitoring history collected in this situation is known as "suspension history". It is unable to assign an accurate RUL value to this type of data used for model training.



Ident	Date	WorkingAge	HN	Precedence	Event	P1H_Par1	P1H_Par2	P1H_Par3	P1H_Par4	P1H_Par5	P1V_Gs	P1V_Overall	P1V_Par1	P1V_Par2	P1V_Par3	P1V_Par4	P1V_Par5
32-4165	14-Feb-01	36936	1	10	B	0	0	0	0	0	0	0	0	0	0	0	0
32-4165	15-Feb-01	36937	1	0	*	0.04	0.043	0.068	0.048	0.035	0.381	0.067	0.047	0.027	0.022	0.027	0.02
32-4165	2-Apr-01	36983	1	0	*	0.038	0.032	0.064	0.049	0.03	0.373	0.067	0.049	0.024	0.027	0.022	0.018
32-4165	26-Apr-01	37007	1	0	*	0.036	0.037	0.07	0.043	0.033	0.4	0.063	0.037	0.022	0.03	0.026	0.02
32-4165	13-Jun-01	37055	1	0	*	0.042	0.037	0.079	0.057	0.036	0.363	0.072	0.052	0.026	0.023	0.029	0.021
32-4165	11-Jul-01	37083	1	0	*	0.045	0.042	0.096	0.056	0.038	0.498	0.073	0.049	0.023	0.034	0.029	0.022
32-4165	20-Aug-01	37123	1	0	*	0.048	0.034	0.077	0.07	0.038	0.582	0.079	0.061	0.026	0.023	0.028	0.02
32-4165	25-Sep-01	37159	1	0	*	0.038	0.034	0.076	0.064	0.038	0.883	0.068	0.042	0.024	0.019	0.033	0.025
32-4165	17-Oct-01	37181	1	0	*	0.031	0.041	0.069	0.062	0.041	0.781	0.072	0.04	0.025	0.029	0.035	0.027
32-4165	27-Nov-01	37222	1	0	*	0.04	0.04	0.077	0.053	0.045	1.047	0.08	0.044	0.021	0.041	0.03	0.03
32-4165	11-Jan-02	37267	1	0	*	0.043	0.034	0.088	0.061	0.05	0.746	0.073	0.04	0.027	0.026	0.03	0.033
32-4165	7-Feb-02	37294	1	0	*	0.077	0.036	0.081	0.054	0.078	2.031	0.1	0.051	0.027	0.025	0.038	0.064
32-4165	11-Mar-02	37326	1	0	*	0.08	0.042	0.095	0.085	0.09	1.836	0.121	0.076	0.029	0.023	0.063	0.059
32-4165	12-Mar-02	37327	1	0	*	0.082	0.035	0.076	0.085	0.089	1.898	0.124	0.079	0.027	0.023	0.064	0.062
32-4165	15-Mar-02	37330	1	0	*	0.043	0.046	0.105	0.082	0.098	1.992	0.119	0.037	0.029	0.034	0.076	0.062
32-4165	3-Apr-02	37349	1	0	*	0.077	0.05	0.071	0.119	0.097	1.516	0.151	0.077	0.026	0.028	0.095	0.081
32-4165	5-Apr-02	37351	1	0	*	0.074	0.028	0.066	0.123	0.094	1.445	0.165	0.077	0.023	0.031	0.098	0.1
32-4165	21-May-02	37397	1	0	*	0.045	0.041	0.082	0.21	0.09	1.688	0.21	0.039	0.022	0.03	0.175	0.087
32-4165	28-May-02	37404	1	0	*	0.041	0.047	0.07	0.311	0.1	2.609	0.263	0.035	0.023	0.019	0.24	0.085
32-4165	2-Jun-02	37409	1	8	EF	0.041	0.047	0.07	0.311	0.1	2.609	0.263	0.035	0.023	0.019	0.24	0.085

Fig. 2.1: Illustration of historical data used for prognostics

Most of the methods utilized to predict the RUL of machines rely on having a substantial number of failure histories. These failure histories provide information on when a machine has experienced a failure and has been replaced. By analyzing this failure information, RUL values can be assigned to different sets of monitored features in the failure history, which are then used to train the prognostic model. However, in real-world scenarios, engineering assets are often taken out of service before experiencing complete failures, either for planned maintenance or when an early defect is detected. As a result, only a limited number of failure histories are available, while suspension histories can be collected more frequently, often more than twice as many. In this situation, the insufficient amount of failure history data makes it challenging to accurately understand the degradation patterns of the machine using a prognostic model. To overcome this limitation, it is crucial to incorporate suspension history into the prognostic model to improve the accuracy of RUL predictions.

Some methods have been reported to incorporate suspension histories in the development of models. These methods have been discussed in detail in Section 1.2.1. They can be categorized into three main groups. The first category is the reliability-based method, which is relatively straightforward to implement. This approach utilizes the failure time and suspension time to estimate the distribution of lifetime, as mentioned in [23]-[26]. However, it is noted that these methods may not be able to predict RUL based on data monitoring. The second group of methods, as presented in [27]-[30], involves establishing a health indicator (HI) for each suspension history. A threshold is set to determine when failure occurs, and this allows for the estimation of failure

time for each suspension history. The suspension history can then be labeled with RUL values for training prognostic models. One drawback of this method is that the estimated failure time is highly dependent on a suitable threshold, which requires a large amount of data or knowledge. This means that this method can only perform well under conditions that are fully documented in the training dataset. The third category of techniques focuses on incorporating both failure and suspension histories in semi-supervised learning. In these methods, failure and suspension histories are treated as labeled and unlabeled data, respectively, as illustrated in [34]-[37]. However, a limitation of these approaches is that the predicted RUL for all inspection points in a suspension history is not interdependent. When using semi-supervised learning methods, the suspension history may yield multiple predicted failure times with significant discrepancies when predictions are made at different inspection points. Additionally, the predicted failure time may be shorter than the suspension time when the predictions are made at early inspection points. As a result, these predictions could be incorrect or even physically implausible.

Generative adversarial network (GAN) is a widely used technique when there is a limited amount of labeled data available. It combines the generative algorithm and adversarial concept to enhance the model generalization [94]. In the field of prognostics, semi-supervised regression GAN (SRGAN) [95] allows for the use of failure and suspension histories as labeled and unlabeled data to make RUL predictions. Nevertheless, existing studies have overlooked the valuable information provided by suspension histories in deep learning models, resulting in less reliable and convincing prognostics. In this chapter, a novel prognostic method is proposed based on the architecture of SRGAN that takes into account the information from suspension histories. To the best of our knowledge, no previous published research has considered the unique failure information provided by suspension histories in the semi-supervised prognostic model.

The original contributions are summarized as follows: 1) The model generalization is improved by using adversarial training for possible multiple failure modes and operating conditions, especially when the number of failure histories is limited. 2) A single objective function incorporating multiple tasks is proposed for SRGAN to consider the failure information from suspension histories. 3) A robustness evaluation method is proposed to assess the uncertainty of the prognostic model resulting from a scarcity of data.

2.2 Semi-supervised regression GANs for prognostics

In this section, the basic principles of SRGAN used for RUL predictions is presented. Let $D_l = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$ represents m labeled degradation data and assume that there are n unlabeled samples, denoted by $D_u = \{x_{m+1}, x_{m+2}, \ldots, x_{m+n}\}$, where x denotes the measurements and y represents the corresponding RUL. The problem of semi-supervised prognostics can be formulated as: given m labeled data D_l and n unlabeled data D_u, construct a nonlinear prognostic model $\hat{y} = f(x)$ to predict the RUL for new condition measurements.

Recently, Olmschenk et al. [95] proposed a novel SRGAN structure for semi-supervised regression and proved that it could perform better than other GAN models [96]. As shown in Fig. 2.2, the SRGAN-based prognostic model not only relies on labeled condition data for supervised learning but also uses unlabeled samples for unsupervised training. Oftentimes, when the amount of available labeled data is limited, the prognostic model can only learn a rough degradation trace, while the unlabeled condition data can make the learned conditional distribution $p(y \mid x)$ more complete because the condition monitoring data of equipment may have similar degradation manifolds. Different from the basic GAN model, SRGAN constructs continuous loss functions instead of cross-entropy losses so that a conditional distribution $p(y \mid x)$ can be effectively learned for RUL predictions.

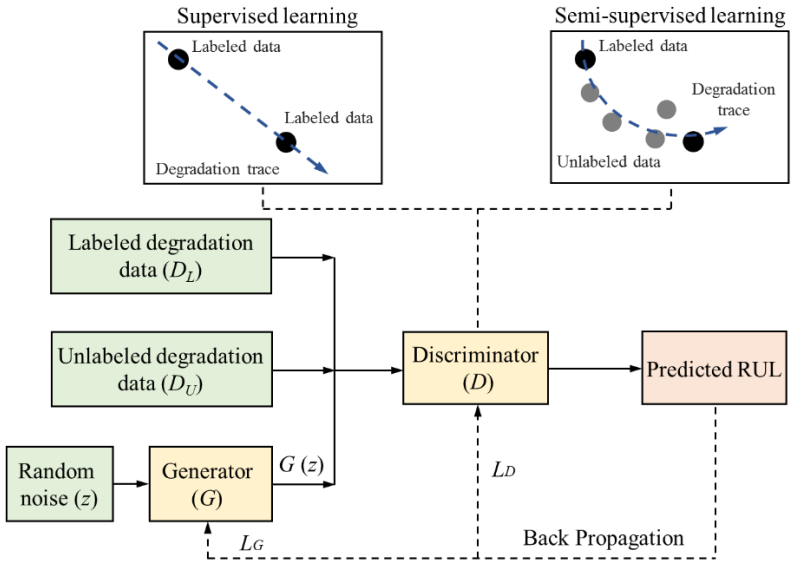


Fig. 2.2: Use of SRGAN for prognostics

According to SRGAN, the discriminator uses the multi-task structure, and its loss function contains three parts as:

$$L_D = L_{supervised} + L_{unsupervised} = L_{labeled} + L_{unlabeled} + L_{generated} \tag{2.1}$$

where $L_{labeled}$ means the loss function for the supervised training task, $L_{unlabeled}$ means estimating the continuous label of unlabeled data, and $L_{generated}$ means judging the input data as real data or a generated one. Each of the loss functions can be defined as follows:

$$L_{labeled} = \mathbb{E}_{x,y\sim pdata(x,y)}\left[D(x)-y\right]^2 \tag{2.2}$$

$$L_{unlabeled} = \left\|\mathbb{E}_{x\sim p_{labeled}} f(x) - \mathbb{E}_{x\sim p_{unlabeled}} f(x)\right\|_2^2 \tag{2.3}$$

$$L_{generated} = -\left\|\log\left(\left|\mathbb{E}_{x\sim p_g} f(x) - \mathbb{E}_{x\sim p_{unlabeled}} f(x)\right|\right)\right\|_1 \tag{2.4}$$

where $\|\bullet\|_1$ and $\|\bullet\|_2$ denote the first-order norm and the second-order norm, $f(x)$ is the activations on an intermediate layer of the discriminator, $L_{labeled}$ is defined by mean squared error (MSE) for the supervised regression, $L_{unlabeled}$ is constructed by the feature matching [97] to reduce the difference in feature statistics between labeled and unlabeled data, and $L_{generated}$ is built by the feature contrasting [95] to make the feature distribution of the real data as dissimilar to the generated data as possible.

The generator also uses feature matching to model the divergence between the real data and the generated data by:

$$L_G = \left\|\mathbb{E}_{x\sim p_g} f(x) - \mathbb{E}_{x\sim p_{unlabeled}} f(x)\right\|_2^2 \tag{2.5}$$

In SRGAN, the generator loss L_G and $L_{generated}$ have opposing goals and thus can form adversarial training to boost the generalization of the prognostic model. For actual RUL prediction applications, the condition monitoring data often continues throughout the life cycle of the equipment. When an SRGAN-based prognostic model is constructed, the failure history and the suspension history will be considered as labeled data and unlabeled data for model training so that

the suspension history can be used to help the prognostic model find a clear degradation trace even with reduced amounts of failure history.

2.3 Proposed methodology

2.3.1 Overview

In most semi-supervised learning-based prognostic methods, the suspension histories are treated as unlabeled data, and the estimated failure time of the suspension histories strongly depends on the real failure histories. However, the suspension histories consist of valuable information that can reflect the degradation of mechanical systems. The special information of suspended data can help the model get more accurate results and cannot be ignored. A suspension history will contain a few inspection points and a termination point at the suspension time, and a suspension history will be labeled by one identity number, such as the first or the second suspension history. Thus, from the perspective of RUL prediction, a suspension history has three special information labels, i.e., identity, suspension time, and inspection time. Based on these three special labels, three valuable pieces of information can be summarized associated with a suspension history:

1) The similar failure time: each of the inspection points in one suspension history should have a close predicted failure time. Due to the measurement noise, the predicted failure time made at all inspection points will not be precisely the same, but they should follow a specific distribution. Therefore, we can assume that the predicted failure time obeys a normal distribution and reduces the difference between all values by minimizing the predicted variance.
2) The suspension time as a model constraint: the associated components were taken out of service at the suspension times, and they did not experience real failures. This information can be used as a constraint for the RUL prediction model. When a prediction model is applied to a suspension history, the predicted failure time for all inspection points should be ensured to exceed the suspension time.
3) The degradation is reflected in the condition monitoring data: the equipment will degrade over time, and the condition monitoring data will change as the degradation progresses. Therefore, the monitoring measurements can reflect the degradation level of equipment

and RUL. Almost all RUL prediction models suppose that the condition data with similar measurements will have similar RUL, which is also applicable for suspension histories.

The above-mentioned three pieces of important information on suspension histories can help the RUL prediction model develop multiple parts of the objective function to get better prediction performance. It is worth mentioning that the third piece of information can also be regarded as the manifold hypothesis, and thus, the suspended data can be highly beneficial for semi-supervised learning.

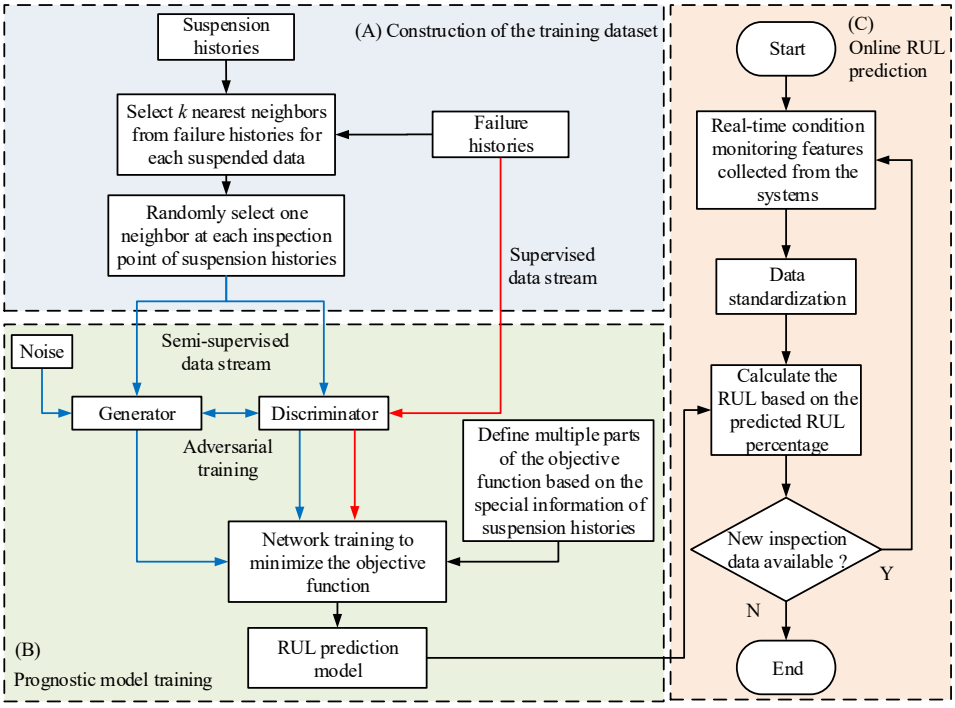


Fig. 2.3: Procedure of RUL prediction using failure and suspension histories

Fig. 2.3 summarizes the significant steps of the proposed semi-supervised GAN method for RUL prediction by integrating failure and suspension histories. The model is trained offline and is applied for online prediction. In order to meet the manifold hypothesis, first in part (A), it is necessary to select those condition data from the failure histories at each inspection point that match suspension histories the best. After that, in part (B), the RUL prediction model will be trained based on the failure and suspension histories by defining multiple parts of the objective function. Once the prediction model is trained, it can be used for RUL prediction online in part (C) when new condition data are available. The red line shows the data stream in the supervised

training step, and the blue line presents the input data for the semi-supervised training step. The most challenging part of using data from suspension histories is to estimate the unknown single failure time at each inspection point. However, unlike the previous works, the key idea of the present work is that a pair of condition data with the same RUL should have a similar distribution instead of independently predicting the failure time of suspension histories directly based on the training model. By using the generative adversarial architecture in semi-supervised training, the predicted failure time of suspension histories will utilize both generative data and failure histories, thereby improving the model generalization and avoiding the overfitting problem.

The details of the proposed method are introduced in the following subsections.

2.3.2 Data preprocessing

When failure histories and suspension histories are available, the first step of the method is to convert the collected data into the samples used for the model training. Let $D_f = \{D_{f,1}, D_{f,2}, ..., D_{f,m}\}$ represent m failure data, where $D_{f,i}$ is the i-th failure history with lf_i inspection points, denoted by $D_{f,i} = \{(x_{f,i,1}, t_{f,i,1}), (x_{f,i,2}, t_{f,i,2}), ..., (x_{f,i,lfi}, t_{f,i,lfi})\}$, and $(x_{f,i,j}, t_{f,i,j})$ is the j-th inspection point of the i-th failure history, where $x_{f,i,j} \in R^{1\times d}$ has d condition monitoring measurement values and $t_{f,i,j}$ is the corresponding inspection time. In addition, suppose that there are n suspension data, denoted by $D_s = \{D_{s,1}, D_{s,2}, ..., D_{s,n}\}$, and often $n > m$. Similar to the failure data, each suspended data has ls_i inspection points, denoted by $D_{s,i} = \{(x_{s,i,1}, t_{s,i,1}), (x_{s,i,2}, t_{s,i,2}), ..., (x_{s,i,lsi}, t_{s,i,lsi})\}$, where $(x_{s,i,j}, t_{s,i,j})$ is the j-th inspection point of the i-th suspension history, and the measurement types of $x_{s,i,j}$ are the same as the failure histories. We call the last inspection point of the i-th failure history $(x_{f,i,lfi}, t_{f,i,lfi})$ as the failure point, and also, $t_{f,i,lfi}$ can be regarded as the failure time of the i-th failure history, represented as $T_{f,i}$, $i = 1, 2, \ldots, m$. In contrast, the last inspection time of the i-th suspension history $t_{s,i,lsi}$ is called the suspension time, denoted by $T_{s,i}$, $i = 1, 2, \ldots, n$.

In this work, the architecture and loss functions of the SRGAN-based prognostic model presented in Section 2.2 are developed. The inputs to the discriminator of the SRGAN model are set as the d condition monitoring measurements collected at each inspection point. The output of the discriminator is the predicted RUL percentage, which is equal to the quotient of the predicted RUL and the failure time. Therefore, the actual RUL percentage should be calculated for all inspection

points of failure histories. For instance, in terms of the inspection data $x_{f,i,j}$ in the i-th failure history, the corresponding actual RUL percentage $r_{i,j}$ will be:

$$r_{i,j} = \frac{T_{f,i} - t_{f,i,j}}{T_{f,i}} \tag{2.6}$$

The training dataset of the proposed method contains labeled data and unlabeled data. The labeled data is constructed based on the failure histories, presented by $D_l = \{(x_{f,1,1}, r_{1,1}), \ldots, (x_{f,1,lf1}, r_{1,lf1}), \ldots, (x_{f,m,1}, r_{m,1}), \ldots, (x_{f,m,lfm}, r_{m,lfm})\}$. In contrast, the unlabeled data will be built based on the suspension histories. Due to lack of failure time for such data, it is hard to estimate the RUL percentage for each inspection point in suspension histories. As mentioned in Section 2.3.1, however, three special labels are critical for suspension histories, i.e., identity, suspension time, and inspection time. For the j-th inspection point of the i-th suspension history $x_{s,i,j}$, the identity, suspension time, and inspection time will be i, $T_{s,i}$, and $t_{s,i,j}$, respectively. From the perspective of RUL prediction, the data with similar monitoring measurements will have similar RUL, while the measurement distribution between the data with different RULs may be completely different. Therefore, the k-nearest neighbor [98] is used to construct the neighbor graph G for suspension histories in data preprocessing. We calculate the k closest neighbors in failure histories D_l for each inspection point in suspension histories, namely $\Omega_{i,j}$, $i = 1, 2, \ldots, n, j = 1, 2, \ldots, ls_i$. The Euclidean distance is used to measure the dissimilarity between a pair of suspended data and failure data as:

$$Dis\left(x_s, x_f\right) = \sqrt{\sum_{i=1}^{d}\left(x_s^{(i)} - x_f^{(i)}\right)^2} \tag{2.7}$$

Here, x_f and x_s denote the condition monitoring measurements in the failure and suspension histories, separately. For each suspended data $x_{s,i,j}$, the Euclidean distances between $x_{s,i,j}$ and all inspection points in failure histories are calculated, and the k failure data with the smallest dissimilarity are recorded in $\Omega_{i,j}$. In the case studies, we found that using different distances will not have a significant influence on the RUL prediction. For each semi-supervised training step, among these k nearest neighbors, only one of them will be selected randomly as the single neighbor of each suspension history data at each inspection point.

Finally, the unlabeled data will be constructed for the model training as $D_u = \{(x_{s,1,1}, 1, T_{s,1}, t_{s,1,1}, \Omega_{1,1}), \ldots, (x_{s,1,ls1}, ls_1, T_{s,ls1}, t_{s,1,ls1}, \Omega_{1,ls1}), \ldots, (x_{s,n,1}, n, T_{s,n}, t_{s,n,1}, \Omega_{n,1}), \ldots, (x_{s,n,lsn}, ls_n, T_{s,lsn}, t_{s,n,lsn}, \Omega_{n,lsn})\}$. In general, 2/3 labeled data and all unlabeled data are used for the model training, and the rest of 1/3 labeled data can be used for the model validation. The input data can also involve multiple time points when the considered length of the time window is larger than one.

2.3.3 RUL prediction considering suspension histories

In this section, SRGAN is developed for RUL prediction based on the failure and suspension histories. As discussed before, the suspended data is especially helpful for the semi-supervised regression due to the manifold hypothesis. In addition, suspension histories often have special labels, and these labels can provide more information than unlabeled data for semi-supervised learning. Therefore, three pieces of valuable information on suspension histories are considered to develop the objective functions of the model to get more accurate prediction results.

The key idea is that the degradation relationship between the condition monitoring data and RUL is the same for all failure and suspension histories. Consequently, for a specific RUL value, the estimated distribution of measurements should be similar. In this method, the discriminator does not attempt to predict the RUL percentage for suspension histories. Instead, the statistics of monitoring measurements with a specific RUL for failure and suspension histories are compared.

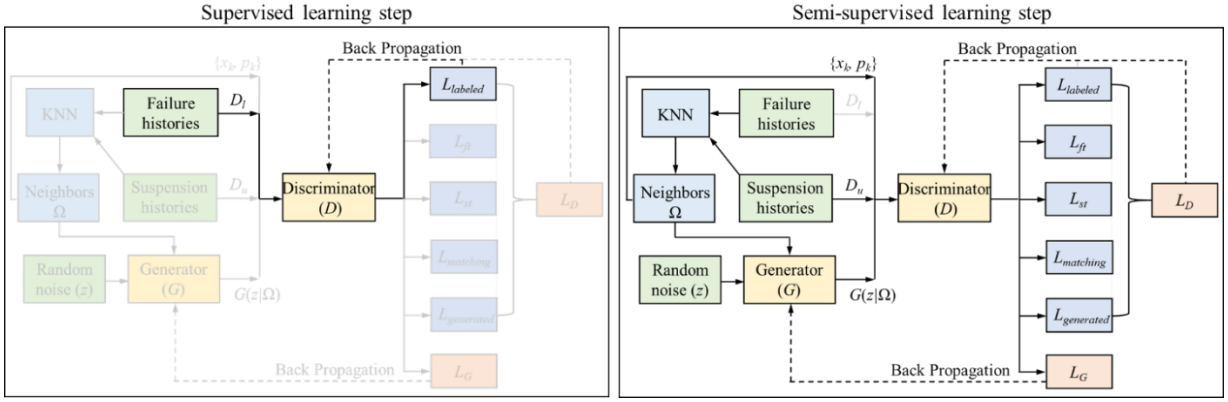


Fig. 2.4: Architecture of the proposed semi-supervised GAN model for RUL prediction

Due to the unknown failure time, only one failure data, denoted by $(x_{k,i,j}, r_{k,i,j})$, is selected randomly from the neighbor set $\Omega_{i,j}$ including k neighbors for the suspended data $x_{s,i,j}$ first, $i = 1, 2, \ldots, n, j = 1, 2, \ldots, ls_i$. We use the discriminator to make the suspended input data $x_{s,i,j}$ have a similar

measurement distribution as the corresponding failure sample $x_{k,i,j}$. The discriminator also tries to make the data generated under the $r_{k,i,j}$ condition have a different measurement distribution from the suspended data $x_{s,i,j}$. On the other hand, the generator seeks to generate samples that match the suspended data $x_{s,i,j}$ under the given $r_{k,i,j}$. Therefore, the objective of the model can be formulated as a two-player non-cooperative game. The architecture of the proposed RUL prediction model is shown in Fig. 2.4. At each epoch during training, all failure histories are first used to update the weights of the discriminator to minimize the supervised regression loss. After that, the discriminator and the generator will be optimized again by using a semi-supervised multi-task structure. Meanwhile, the generator applies a conditional structure to produce samples to improve the model generalization. Similar to SRGAN, we also use feature matching and feature contrasting to describe the dissimilarity between two distributions. In these settings, the predicted RUL is still determined by the failure histories, but the fact that the model must meet the special information and must match the suspension history with the failure history one-to-one can force the discriminator to obtain more accurate RUL predictions.

Primarily in semi-supervised training, the loss of discriminator is constructed as a multi-task function, which contains five parts to represent the useful information of suspension histories as:

$$L_D = \left\{ L_{labeled}, L_{unlabeled}, L_{generated} \right\} = \left\{ L_{labeled}, L_{ft}, L_{st}, L_{matching}, L_{generated} \right\} \tag{2.8}$$

Given the labeled data D_l, the $L_{labeled}$ is defined by MSE for supervised regression as:

$$L_{labeled} = \mathbb{E}_{x_f, p \sim p_{data}(x_f, p)} \left[D(x_f) - r \right]^2 \tag{2.9}$$

where $p_{data}(x_f, r)$ is the joint probability distribution of the condition monitoring measurements and the corresponding RUL percentage in D_l.

The loss L_{ft} is built based on the first and the third label, i.e., the identity and inspection time, of suspension histories, which means minimizing the difference of the predicted failure time among all inspection points in one suspension history by:

$$L_{ft} = \frac{1}{n} \sum_{i=1}^{n} L_{ft,i} = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{ls_i} \frac{1}{ls_i} \left[\frac{1}{ls_i} \sum_{j=1}^{ls_i} T^*_{f,i,j} - T^*_{f,i,j} \right]^2, \text{ with } T^*_{f,i,j} = \frac{t_{s,i,j}}{T_{s,i}(1 - D(x_{s,i,j}))} \tag{2.10}$$

where n is the number of suspension histories and $D(x_{s,i,j})$ denotes the output of discriminator with $x_{s,i,j}$ as input.

Since the predicted failure time of suspension histories should exceed their suspension time, the loss L_{st} is set as a constraint for the RUL prediction model by:

$$L_{st} = \frac{1}{n}\sum_{i=1}^{n} L_{st,i} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{ls_i}\frac{1}{ls_i}\left[1 - t^*_{s,i,j}\right]^2 \text{, with } t^*_{s,i,j} = \begin{cases} T^*_{f,i,j}, & if\ T^*_{f,i,j} \le 1 \\ 1, & if\ T^*_{f,i,j} > 1 \end{cases} \tag{2.11}$$

Finally, the loss $L_{matching}$ means minimizing the difference in measurement statistics between unlabeled data D_u and their corresponding one random neighbor (x_k, r_k), and $L_{generated}$ means maximizing the distribution divergence between the generated and unlabeled data given the specific condition of r_k. Same as the losses constructed in SRGAN, i.e., Eq. (2.3) and Eq. (2.4), the loss $L_{matching}$ and $L_{generated}$ are still built by feature matching (measurement matching) and feature contrasting (measurement contrasting), respectively, but using conditional formulas:

$$L_{matching} = \left\| \mathbb{E}_{x_k \sim p(\Omega|x_s)} f(x_k) - \mathbb{E}_{x_s \sim p_{unlabeled}} f(x_s) \right\|_2^2 \tag{2.12}$$

$$L_{generated} = -\left\| \log\left(\left| \mathbb{E}_{z \sim noise, p_k \sim p(\Omega|x_s)} f(G(z \mid p_k)) - \mathbb{E}_{x_s \sim p_{unlabeled}} f(x_s) \right| \right) \right\|_1 \tag{2.13}$$

where $p_{unlabeled}$ is the probability distribution of the monitoring measurements x_s in suspended data D_u, and $p(\Omega|x_s)$ is the conditional probability distribution of the neighbor set given a specific x_s.

Simultaneously, the loss of the generator is also constructed as a conditional form by matching different measurements:

$$L_G = \left\| \mathbb{E}_{z \sim noise, p_k \sim p(\Omega|x_s)} f(G(z \mid r_k)) - \mathbb{E}_{x_s \sim p_{unlabeled}} f(x_s) \right\|_2^2 \tag{2.14}$$

Often, the objective function of machine learning algorithms can be decomposed into a summation of the training samples. Therefore, we can decompose the training data into multiple mini-batch samples and perform model training separately. When the training data D_l and D_u are available, the losses proposed from Eq. (2.8) to Eq. (2.14) can be used to optimize the RUL prediction model in several epochs. In each epoch, the labeled data D_l is first used for the supervised training. After that, the i-th suspended data with a batch size of ls_i and their corresponding neighbors x_k are put

into the model to minimize the losses $L_{labeled}$, $L_{ft,i}$, $L_{st,i}$, $L_{matching}$, $L_{generated}$, and L_G successively for i from 0 to n. When the training epoch reaches the maximum preset epoch, the model training is completed, and the discriminator with the minimum validation loss is recorded for the online RUL prediction.

To summarize, the proposed method sets the objective function based on the special information of suspension histories. It seems that only using a common regression model with the proposed loss functions may still be feasible. While considering that the data set usually contains noise, the use of adversarial training can help the model improve the quality of generated data to reduce overfitting based on suspension histories. Consequently, the proposed RUL prediction model can be especially useful when the amount of failure history is limited. In actual applications, it is reasonable to estimate RUL as a constant value when the equipment operates in normal conditions. When the maximum RUL is considered, we can first calculate the value of ($T_{s,i}$ – $t_{s,i,j}$) for each suspended data $x_{s,i,j}$. If the value is larger than the preset maximum RUL, the corresponding suspended data can be regarded as failure data with the RUL percentage equal to 1, otherwise, $T^*_{f,i,j}$ and $t^*_{s,i,j}$ are estimated for the semi-supervised training.

2.4 Model analysis

In this section, we have an in-depth analysis of the proposed RUL prediction method in order to better implement the model. In Section 2.4.1, the training mechanisms and model setting are first discussed. Then in the rest of the section, the model uncertainty is analyzed to improve the model credibility.

2.4.1 Model theorem and parameter setting

In the proposed RUL prediction model, multiple tasks are designed based on the information of suspension histories to improve the model performance. Given the loss functions listed from Eq. (2.9) to Eq. (2.13), the objective function of the discriminator can also be formulated as:

$$\min_D V_D(D,G) = \lambda_1 L_{labeled} + \lambda_2 L_{ft} + \lambda_3 L_{st} + \lambda_4 L_{matching} + \lambda_5 L_{generated} \tag{2.15}$$

where λ_1, λ_2, λ_3, λ_4, and λ_5 denote the weight of the five tasks in $\min_D V_D(D, G)$.

In the discriminator, the parameters θ_d of hidden layers are shared between all tasks. When a generator is trained properly, each pair of the distribution between the labeled, unlabeled, and generated data will be as similar as possible. For this fixed generator, we can find an optimal discriminator D_G^* by minimizing the loss function. When conducting model training on supervised learning, we aim to learn a good representation for RUL prediction that ideally ignores the data-dependent noise in the failure histories. As suspended data have different noise patterns, a model that considers two sources of equipment histories can learn a more general representation. In this method, the weight of loss can be adjusted to help the model focus its attention on the critical measurements of failure histories, and the extra training tasks will provide additional evidence for the relevance or irrelevance of those measurements. Therefore, the unsupervised loss designed by the special information of suspension histories can be regarded as a regularizer for the supervised loss to reduce the risk of overfitting [99].

The weight of the loss determines the importance of the corresponding optimization task. For example, if the λ_1 parameter is much larger than other weight parameters, the proposed model will become a purely supervised learning model that uses only failure histories. In this work, the weight setting is determined using the following approach: we first set the weight of semi-supervised tasks to ensure that the losses of all training tasks are in the same magnitude order. Then, we change the value of λ_1 many times to train the model and select the final λ_1 with the smallest validation loss. This procedure does not lead to the optimal weighting factors, but if the model is well constructed and all of the losses are in the same magnitude order, the adjustment of a single weighting parameter will not significantly change the prognostic performance. As a result, this weight-setting procedure is reasonable to find a set of weights for the proposed prognostic model. In addition to the parameter setting, we can also select different basic models, e.g., multilayer perceptron (MLP), LSTM, or CNN, to construct the RUL prediction model, which depends totally on the type of condition monitoring features [100].

2.4.2 Uncertainty quantification

In prognostics, especially when the failure history is limited, it is of paramount importance to take into account uncertainties in the data and in the prediction models [32]. Consider the labeled failure data, $D_l = \{(x_{f,1,1}, r_{1,1}), \ldots, (x_{f,1,lf1}, r_{1,lf1}), \ldots, (x_{f,m,1}, r_{m,1}), \ldots, (x_{f,m,lfm}, r_{m,lfm})\}$, constructed in Section 2.3.2. In this work, our targets, the RUL percentage of the inspection data $x_{f,i,j}$ in the i-th failure

history, denoted by $r_{i,j}$, are assumed to be drawn from a Gaussian distribution with unknown mean $\mu_{i,j}$ and variance $\sigma_{i,j}^2$. The RUL percentage uncertainty is mainly caused by the variations in the histories for different units, the uncertainty in the training data and the capability of the prediction model. Gaussian distribution is often times used to describe such uncertainty in many prognostic works [101][102], including uncertainty in model parameters. Based on the idea from [103], the value of $\mu_{i,j}$ and $\sigma_{i,j}^2$ can be estimated based on their conjugate prior distribution as:

$$r_{i,j} \sim N\left(\mu_{i,j}, \sigma_{i,j}^2\right), \text{ with } \mu_{i,j} \sim N\left(\gamma_{i,j}, \sigma_{i,j}^2 v_{i,j}^{-1}\right), \sigma_{i,j}^2 \sim \Gamma^{-1}\left(\alpha_{i,j}, \beta_{i,j}\right) \tag{2.16}$$

where $\Gamma(\cdot)$ is the gamma function, $\gamma_{i,j}$ is the mean of $\mu_{i,j}$, $\sigma_{i,j}^2 v_{i,j}^{-1}$ is the variance of $\mu_{i,j}$, $\alpha_{i,j}$ and $\beta_{i,j}$ are the shape and scale parameter of the gamma function, and in which $1 > \gamma_{i,j} > 0$, $v_{i,j} > 0$, $\alpha_{i,j} > 1$, and $\beta_{i,j} > 0$. By these definitions, four values, i.e., γ, v, α, and β, can be set as the output of the discriminator. Considering uncertainties, we aim to learn a prognostic model $\hat{\gamma}, \hat{v}, \hat{\alpha}, \hat{\beta} = f(x)$, so that the distribution of RUL percentage, i.e., $N(\mathbb{E}(\mu), \mathbb{E}(\sigma^2))$, can be predicted when new condition data x is monitored. In order to learn the defined prognostic model, the "labeled loss" in the discriminator can be adjusted as an evidential regression form [103]:

$$L_{labeled} = \mathbb{E}_{x_f, p \sim p_{data}(x_f, p)}\left(L_{NLL} + \lambda L_R\right) \tag{2.17}$$

where L_{NLL} aims to maximize the likelihood of observations, L_R means inflating uncertainty when the prediction is wrong, and λ denotes the weight between the two tasks. For each pair of labeled failure data $\{x_{f,i,j}, p_{i,j}\}$, both of the loss functions are calculated as [103]:

$$\begin{aligned} L_{NLL} = & \frac{1}{2}\log\left(\frac{\pi}{v_{i,j}}\right) - \alpha_{i,j}\log\left(\Omega_{i,j}\right) + \left(\alpha_{i,j} + \frac{1}{2}\right)\log\left(\left(r_{i,j} - \gamma_{i,j}\right)^2 v_{i,j} + \Phi_{i,j}\right) \\ & + \log\left(\frac{\Gamma\left(\alpha_{i,j}\right)}{\Gamma\left(\alpha_{i,j} + 0.5\right)}\right) \end{aligned} \tag{2.18}$$

$$L_R = \left\| p_{i,j} - \gamma_{i,j} \right\|_1 \cdot \left(2v_{i,j} + \alpha_{i,j}\right) \tag{2.19}$$

where $\Phi_{i,j} = 2\beta_{i,j}(1 + v_{i,j})$.

When new condition data x is available for online applications, four output values, i.e., $\hat{\gamma},\hat{v},\hat{\alpha},\hat{\beta}$, are obtained to characterize the distribution of RUL percentage. The predicted RUL percentage will be a Gaussian distribution with mean $\mathbb{E}(\mu)$ and variance $\mathbb{E}(\sigma^2)$ as:

$$\mathbb{E}(\mu)=\hat{\gamma},\ \mathbb{E}(\sigma^2)=\frac{\hat{\beta}}{\hat{\alpha}-1} \tag{2.20}$$

where the mean of μ is also regarded as the output of the discriminator when its input is suspension histories for semi-supervised learning. Therefore, the term $D(x_{s,i,j})$ will be computed as $\gamma_{i,j}$ for unsupervised losses in Eq. (2.10) and Eq. (2.11).

Since the number of failure histories is assumed to be limited, it is critical to explain the uncertainty caused by the scarce data. The value of σ^2 quantifies the noise inherent in observations, and the variance of μ accounts for uncertainty associated with the amount of data [104]. As pointed out in [105], if a prognostic model is well constructed with scarce data, there is an emphasis on calculating the estimate with the minimum variance. That is, the variance of the mean of the predicted RUL percentage should be within an acceptable range. As a result, a credibility value (CV) is defined based on the variance of μ to estimate the credibility of network models in the test process by:

$$CV=\sum_{i=1}^{n_t}\mathrm{Var}(\mu_i)=\sum_{i=1}^{n_t}\frac{\beta_i}{v_i(\alpha_i-1)} \tag{2.21}$$

where n_t is the total number of test data. A smaller CV value indicates that the model has lower uncertainty, and thus, the model will be more reliable for RUL predictions.

It is the first time to integrate the idea of evidential regression into the RUL prediction model to estimate the credibility of the model. By this approach, the uncertainty of the RUL prediction model can be quantified in a reasonable way. The distribution of the predicted RUL percentage can also be estimated through a single run of the GAN model without repeated model training and running.

2.5 Case studies

2.5.1 Data description

In the case study, two benchmark datasets are used to validate the RUL prediction performance of the proposed method. The first dataset is the C-MAPSS dataset [106], which was created by NASA to simulate the actual degradation of turbofan engines. As shown in Table 2.1, the C-MAPSS dataset is divided into four subsets according to different fault modes and operation conditions. Each subset provides a training set and a testing set. The training engine operates from normal to failure, and the corresponding RUL of each time cycle is offered. In contrast, the testing engine stops operating prior to failure. The c-MAPSS dataset contains multiple fault modes and operation conditions and is thus suitable for studying the relationship between varying conditions and degradation trends [16].

Table 2.1: C-MAPSS dataset

Subsets	FD001	FD002	FD003	FD004
Training engines	100	260	100	249
Testing engines	100	259	100	248
Fault modes	1	1	2	2
Operation conditions	1	6	1	6

In the second case study, the data collected from bearings on a group of Gould pumps at a Canadian Kraft pulp mill company [107] is utilized. In this dataset, 11 failure histories and 22 suspension histories are available. Vibration data are collected from bearings in eight pump locations. For each history, seven measurements are analyzed with five different vibration frequency bands and the overall vibration reading plus acceleration data. The pump dataset is used to reflect the degradation prediction performance of the proposed model for real vibration data.

Due to the capability to cope with long-term dependencies, LSTM is used as the baseline model in this work. For the comparative study, three additional models are also constructed so that four contrast models are discussed, including LSTM, SRGAN, LSTM-S, and SRGAN-S. In the experimental models, SRGAN is a semi-supervised regression method introduced in Section 2.2.2; LSTM-S is the method considering suspension histories proposed in [22] but using LSTM instead of ANN; SRGAN-S denotes the proposed method that develops SRGAN to consider both failure

and suspension histories. Before the model training, the input condition monitoring measurements were standardized with zero mean and one deviation, but the output is not preprocessed because the output RUL percentage is a number from 0 to 1.

To evaluate the accuracy of the proposed method, the root mean squared error (RMSE) is estimated as:

$$RMSE = \sqrt{\frac{1}{n_t}\sum_{i=1}^{n_t}\left(RUL_{ture} - RUL_{predicted}\right)^2} \tag{2.22}$$

Additionally, the credibility of network models is estimated based on the CV given by Eq. (2.21). In this work, we use RMSE as the first assessment measure. However, the CV should also be within an acceptable range. If the difference in RMSE between the two methods is not significant, we can choose the method with a lower CV. In practice, since the accuracy and credibility matrices are not sensitive to the value of λ in Eq. (2.17), we set the same λ as 1 for all models to compare their RUL prediction performance.

2.5.2 Case study 1: C-MAPSS dataset

Since part of sensor values remain unchanged, 14 sensors, i.e., sensors 2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20, and 21, are selected as the input measurements. In this case, the maximum RUL is considered as a constant value of 130, so the actual RUL percentage will be equal to 1 if the RUL of an inspection point is larger than 130. Suspension histories are generated from training engines, and the suspension time is randomly selected from 10 to 40 cycles in advance of the failure time. The size of the time window is set as 10 to consider time dependencies. We construct the same LSTM structure for four models. The LSTM structure has two hidden layers with 30 and 10 neurons plus dropout (p = 1), and both activations of hidden layers are tanh functions. The number of neurons in the input and output layer of the discriminator is 14 and 1, respectively, corresponding to the measurement dimension and the 1-dimensional RUL percentage output. The generators of SRGAN and SRGAN-S are constructed as a simple two-layer perceptron with tanh activations. The 200-dimensional random noise that obeys the standard normal distribution is used as the input of the generator, and the output dimension of the generator is 140, which is equal to the product of time window size and measurement dimension. The five closest neighbors are calculated in each failure history for each inspection point in the suspension history. Adam

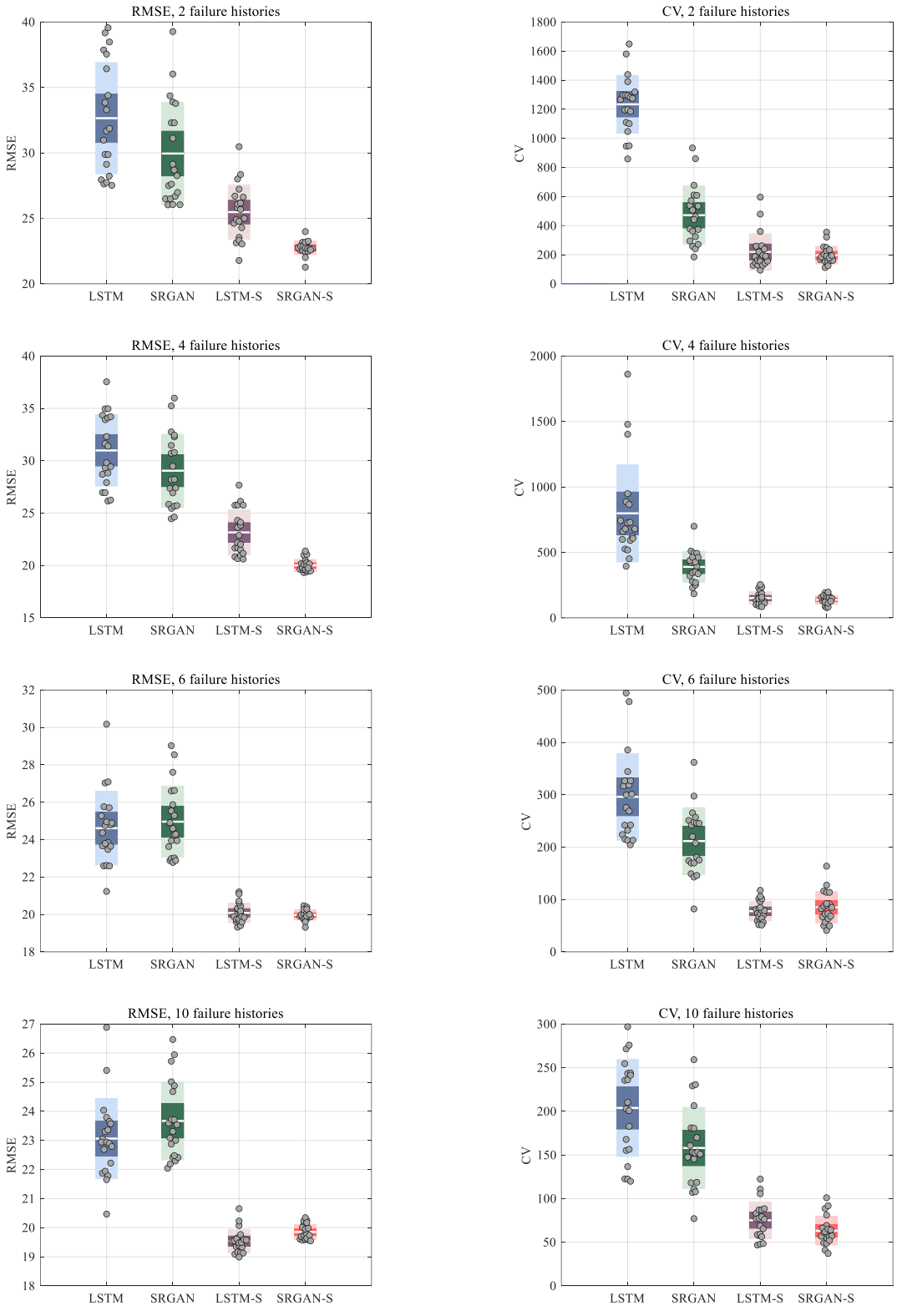


Fig. 2.5: RMSE and CV of different models on FD001 with different failure histories

optimizer is used for the model training with the same 0.5 momentum parameter, while the learning rate and epoch number are determined based on the convergence capability of different models. All testing engines of each subset are used to test the model performance. The RMSE and CV of each model will be recorded with the minimum supervised test loss for the comparison, and therefore, the result can be regarded as the best performance that the model can get.

First, four models are tested using the FD001 subset with ten suspension histories and a different number of failure histories. Subset FD001 is subjected to a single fault mode and operating condition. Fig. 2.5 shows the experimental results of RMSE and CV obtained from 20 simulations using four methods. In Fig. 2.5, the middle line of each box denotes the mean value of 20 tests; the dark zone denotes the 95% confidence interval of the mean value; the low transparency zone represents the region that contains about two-thirds of the data points. Between two boxes, the smaller the overlap of dark bars, or the larger the gap between dark bars, the smaller the statistical P-value and the stronger the evidence for a significant difference between the two methods [108]. It is observed that SRGAN-S outperforms the other three methods with respect to both RMSE and CV, especially when the number of failure histories is very limited, such as only 2 or 4 failure histories. In addition, SRGAN-S shows a statistically significant difference between the RMSE produced by other models. With the increase of failure histories, the predicted failure time of suspension histories will be more accurate, and thus, LSTM-S can get a satisfactory performance in the final subcase. However, when the number of failure histories is 6 or 10, the statistical difference between LSTM-S and SRGAN-S is not significant. This may be due to the fact that the proposed method already shows good performance with only 6 failure histories. Therefore, further increasing the number of failure histories may not lead to a notable improvement in its prognostic performance. From Fig. 2.5, it can be seen that the RMSE variance of the proposed model in 10 failure histories seems to be larger than that in 6 failure histories. This misidentification is mainly caused by the different scales of the y-axis in the two plots. SRGAN-S can get a highly accurate and reliable RUL prediction in nearly all subcases and is thus the most effective method among these four methods when the failure histories are not adequate.

To study the relationship between varying conditions and degradation trends, four models are also verified using different subsets with the same four failure histories and eight suspension histories. Subset FD001 and FD003 are subjected to a single operating condition, while subset FD002 and

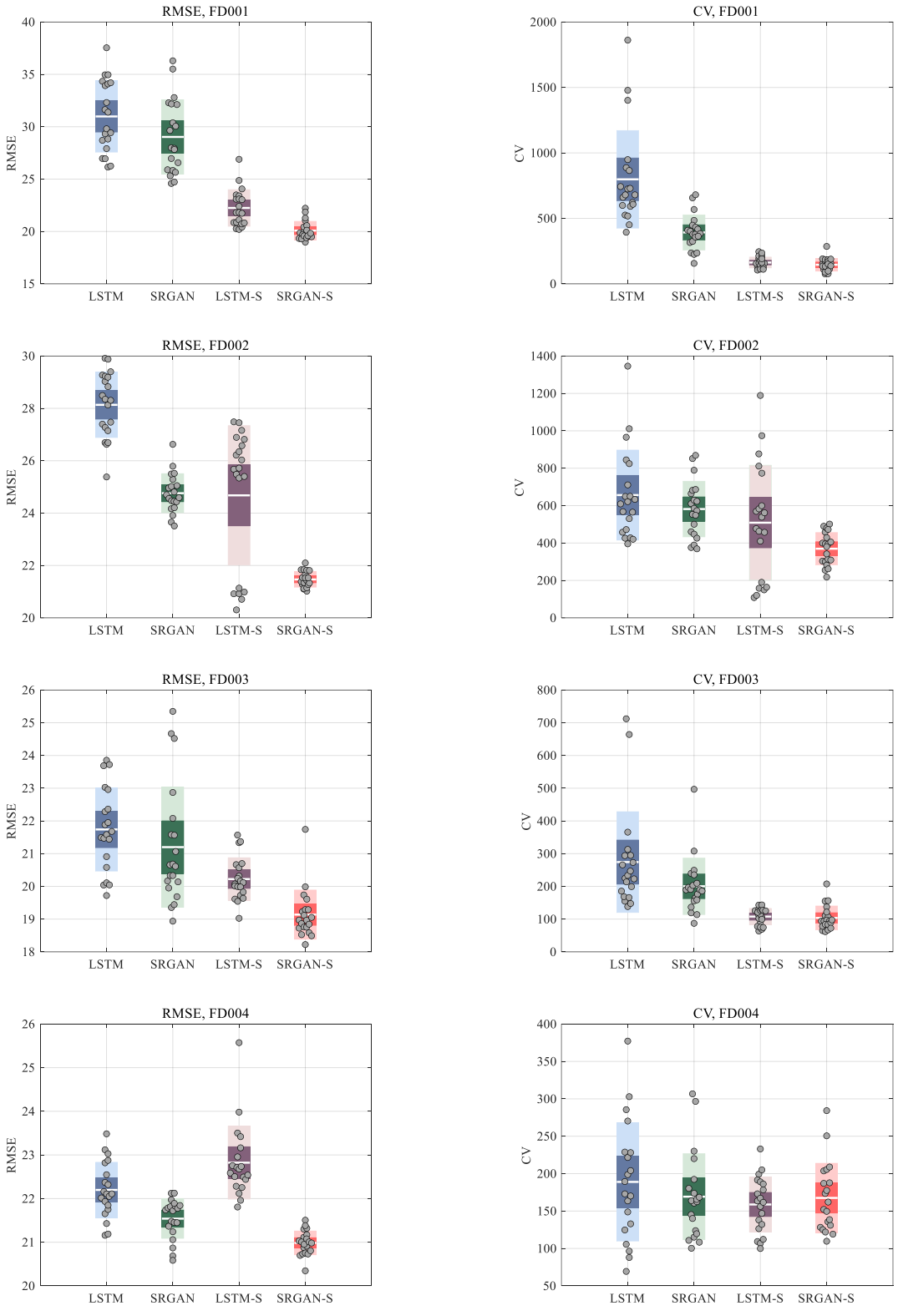


Fig. 2.6: RMSE and CV of different models on different data subsets

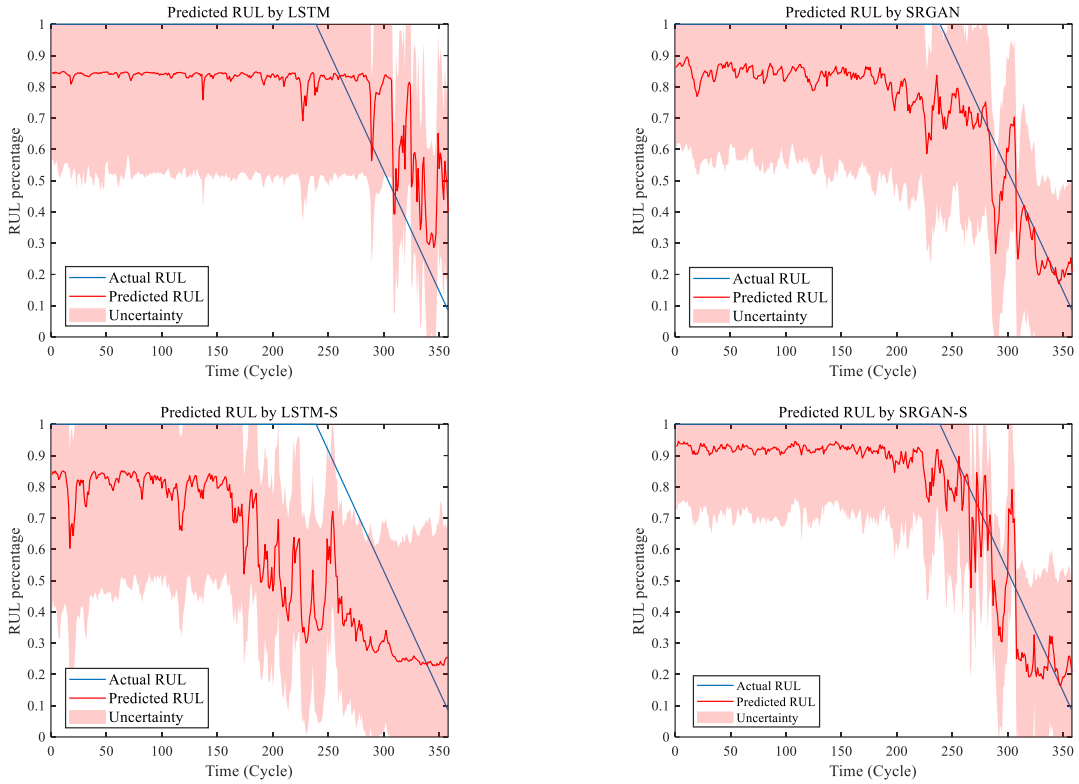


Fig. 2.7: Comparisons between predicted and actual RUL percentages of one testing engine

FD004 are more complex due to six operating conditions. The experimental results of RMSE and CV obtained from 20 simulations are presented in Fig. 2.6. Overall, SRGAN-S performs better than the other three methods because it has the lowest RMSE and a lower acceptable CV in all subcases. Except for FD001, different methods will not show significant differences on the CV for other subsets, indicating that the prediction results of each method are reliable. LSTM-S can get a good performance in single-operation conditions, i.e., FD001 and FD003, but it is hard to obtain a satisfactory performance on FD002 and FD004. If the failure histories cannot contain all conditions, LSTM-S may lead to biased and inconsistent estimates for the failure time of suspension histories, and therefore, the RMSE by LSTM-S may be even worse than the standard LSTM method. In contrast, for SRGAN-S, the failure information of suspension histories will not only rely on the failure histories but also on the generated data, thereby improving the generalization of the model, especially for complex physical conditions.

Fig. 2.7 shows an intuitive comparison between the predicted RUL values and the actual RUL values by different methods. Four models are trained on FD002 with four failure histories and

eight suspension histories. The prediction of the sixty-fifth testing engine is presented as an example for demonstration. By using the uncertainty quantification method proposed in Section 2.4.2, the predicted RUL percentage is a Normal distribution $N(\mu, \sigma^2)$ rather than a single value. In Fig. 2.7, the mean, i.e., red line, and variance of the distribution can be calculated by Eq. (2.20). The uncertainty zone, i.e., light red region, ranges from μ-2σ to μ+2σ, which means there will be more than 95% predicted values in the uncertainty zone. It can be observed that the predicted RUL of SRGAN-S is closer to the actual RUL than the other three methods. In addition, since there will be more degradation information as the failure approaches, the prediction errors of engines in the early operation are larger than those in the later operation. Fig. 2.7 shows great uncertainties in the prediction because the number of failure histories is relatively scarce, and the uncertainty will be significantly reduced when we provide more failure histories.

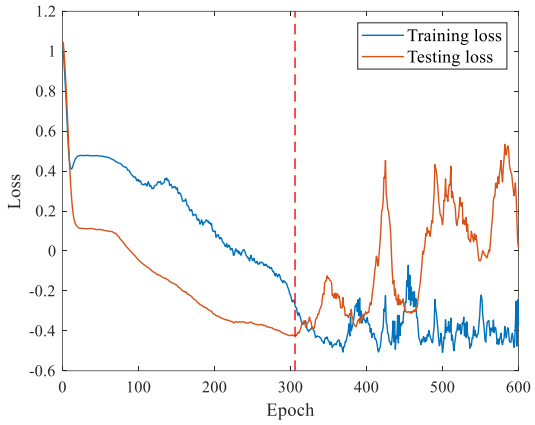


Fig. 2.8: Training loss and testing loss using the proposed method

Table 2.2: Comparison of computational costs by different methods

Methods	LSTM	LSTM-S	SRGAN	SRGAN-S
Offline training time (s / epoch)	7.0×10^{-3}	6.1×10^{-2}	2.1×10^{-2}	2.5×10^{-1}
Online prediction time (s/ 5000 inspections)	4.5×10^{-2}	4.2×10^{-2}	4.2×10^{-2}	4.4×10^{-2}

Fig. 2.8 presents the supervised training loss and testing loss by using the proposed method. At the beginning of training, the testing loss will decrease with the training loss. After a period of epoch, the testing loss and training loss will show obvious fluctuations. As discussed in Section 2.4.1, the unsupervised loss designed by the unique information of suspension histories can be regarded as a regularizer so that SRGAN-S has a strong capability to reduce the risk of overfitting. The regularizer of SRGAN-S improves the training and testing noise when the model is close to

overfitting. Therefore, after the first local minimal point of testing loss, the training loss of SRGAN-S will not continue to decrease, so that a prognostic model with a smaller test loss and better generalization can be found. The computational complexity of different methods is discussed in Table 2.2. Generally, the training of a deep learning model requires hundreds of epochs. For example, we set 300 maximum epochs for LSTM and 600 epochs for the other three models in this case. The proposed SRGAN-S model often requires more offline training time since it not only needs to process all unlabeled data in each epoch but also needs to calculate additional loss functions to consider the information of suspension histories. Nevertheless, all methods use the same LTSM structure for online RUL prediction, and thus, they require almost the same online prognostic time. In addition, for each inspection point, the online prediction time of each model is no more than 9×10^{-6} s, that is, the deep learning models often have high prediction efficiency and low computational cost for online implementation.

2.5.3 Case study 2: Real pump dataset

In the real pump dataset, one measurement has a large amount of missing data. Therefore, six measurements with five vibration frequency bands plus the overall vibration and acceleration are analyzed. Besides, since the inspection time point is discontinuous and has different dispersion, the inspection time is also regarded as a monitoring measure. We can get a total of 43 condition monitoring measurements for the RUL prediction. In this case, we select five failure histories for training and the other four as test histories among 11 available failure histories. In addition, we use ten actual suspension histories, which is twice the number of training failure histories. We set double-time windows for the pump data, and thus, in the training data, we will have 130 failure input/output pairs and 155 suspension input/output pairs. We calculate the two closest neighbors in each failure history for suspended data. Different from Case 1, here, 90 failure data and all suspended data are used for the model training, and the remaining 40 failure data are used for the model validation. In addition, considering that different model architectures may require different network structures, we test several network structures and find the proper LSTM structure for four methods, respectively. Table 2.3 lists the architecture (hidden layers) of four models, where LSTM(n) and FC(n) indicate an LSTM layer and a fully connected layer with n neurons plus dropout (p = 1), separately, and all activations of hidden layers are tanh functions. The RMSE and CV of each model are recorded when the minimum validation loss is reached, which is more

Table 2.3: Model architecture setting

Method	Discriminator architecture	Generator architecture
LSTM	LSTM(30)-LSTM(20)-LSTM(10)	(-)
SRGAN	LSTM(30)-LSTM(20)-LSTM(10)	FC(200)-FC(100)
LSTM-S	LSTM(150)-LSTM(80)-LSTM(50)	(-)
SRGAN-S	LSTM(100)-LSTM(60)-LSTM(20)	FC(200)-FC(100)

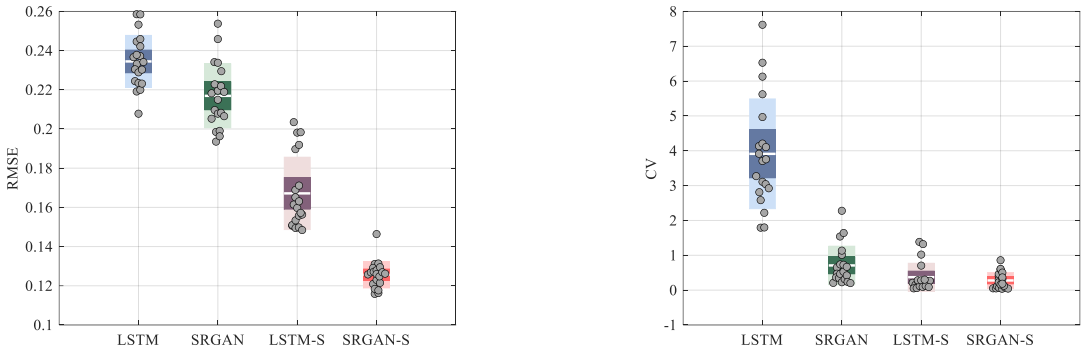


Fig. 2.9: Prognostic performance by different methods on the pump data

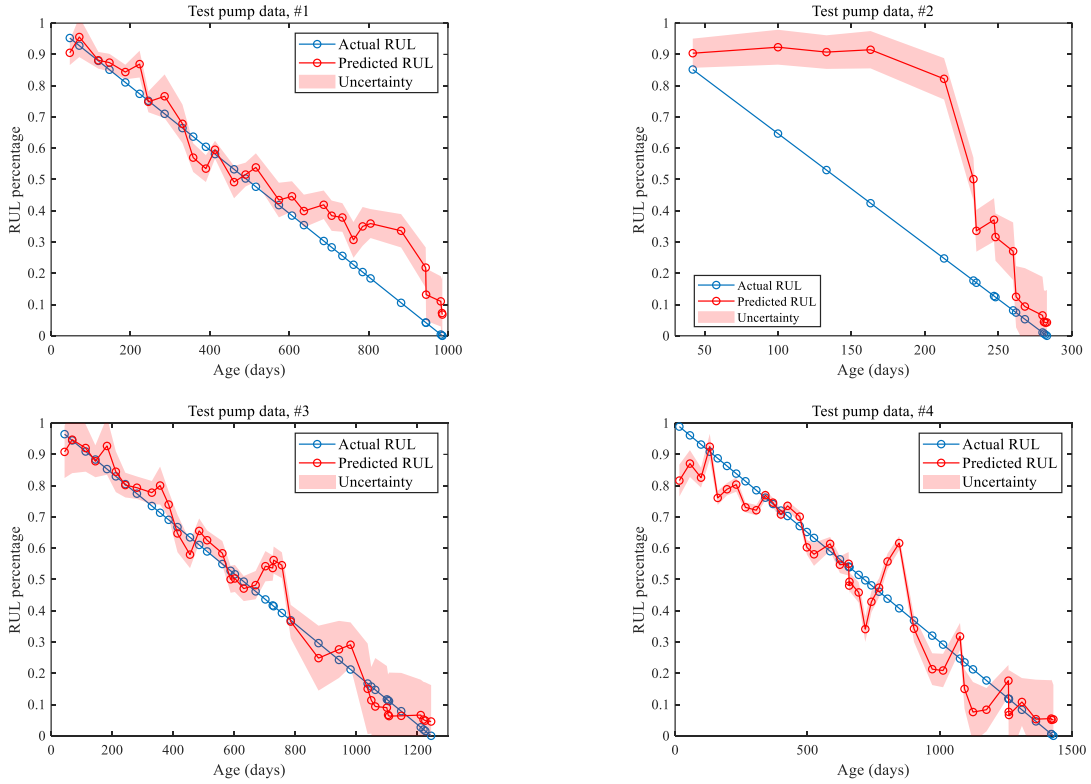


Fig. 2.10: RUL predictions for four tests using the proposed method

practical for the actual applications. Finally, all testing failure histories are used to evaluate the performance of each model.

In this case, RMSE in Eq. (2.22) is computed directly by the predicted and actual RUL percentage. Fig. 2.9 presents the experimental results of RMSE and CV obtained from 20 simulations with different random seeds using four methods. It can be observed that SRGAN-S outperforms the other three methods with respect to both RMSE and CV. In addition, SRGAN-S shows a statistically significant difference in RMSE. LSTM is significantly inferior to other methods in both RMSE and CV, indicating that the supervised deep learning methods are strongly dependent on a large number of failure histories. Except for SRGAN-S, LSTM-S can also get a satisfactory performance, because in this case, the training data can involve almost all conditions in testing data. The CV values of the three methods, SRGAN, LSTM-S, and SRGAN-S, are acceptable and statistically close, which indicates that the prediction results of these three methods are reliable.

The prediction results on four testing histories using SRGAN-S are shown in Fig. 2.10. Overall, the predicted RUL of SRGAN-S is close to the actual RUL. Especially in the second testing history, the pump failed within only 300 days, but the prediction model can also find this abrupt fault and can get a good RUL prediction when the pump is about to fail. The RUL predicted by SRGAN-S has fewer uncertainties in the middle of failure histories, possibly because there are more mid-term inspection points in the training dataset. This case study illustrates the capability of the proposed method to achieve accurate and reliable RUL predictions for the real application.

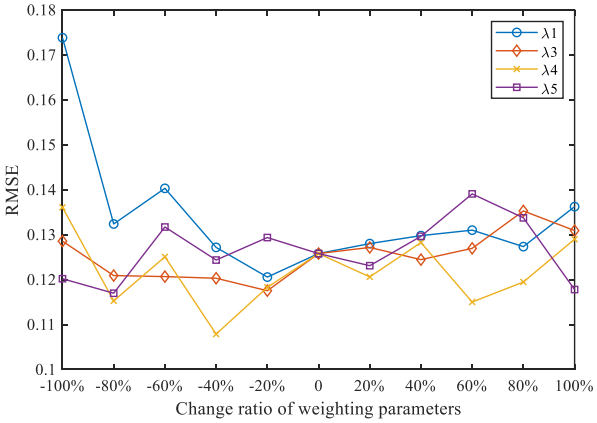


Fig. 2.11: Sensitivity analysis of weighting factors

Finally, Fig. 2.11 provides a sensitivity analysis to analyze how the values of weighting factors affect the prediction results. As shown in Eq. (2.15), we need to set five weighting parameters for the discriminator in the proposed method. Generally, we can only set four weights by setting one of the weights to 1. Assume that the SRGAN-S is well constructed with λ_2 equal to 1, and the results presented in Fig. 2.11 can be obtained. We change the value of each weight parameter from -100% to 100% and keep the remaining weighting parameters fixed to observe the influence of the change of the weight on the test RMSE. It is observed that the prognostic performance results from the combined effect of all weighting factors, and the change of a single parameter will not significantly affect the model performance. In addition, nearly all RMSEs obtained by the proposed method with different weighting parameters are from 0.11 to 0.14, and these results are also better than those of other comparative methods shown in Fig. 2.9.

2.6 Conclusions

This chapter provides a semi-supervised GAN model to predict the RUL by using both failure and suspension histories. Without directly predicting the failure time of each suspension history, the proposed GAN model indirectly estimates the RUL percentage of suspended data by matching the statistical information between similar failure and suspension histories. Meanwhile, the generated data is produced from the generative network to improve the model generalization and avoid overfitting. As a result, the failure information of suspension histories depends not only on the failure histories but also on the generated data. By comparing different prediction methods, the main conclusions from the case studies validate the accuracy and credibility of the proposed method. The proposed GAN model can get satisfactory accuracy, especially when the amount of failure history is limited.

The main contribution of the proposed method is the construction of multiple parts of the objective function to make full use of the information on suspension histories. Specifically, the proposed GAN model considers suspended data as reasonable constraints to improve the model performance, rather than just treating them as unlabeled data. When the equipment has multiple failure modes or operation conditions, using a model built by limited failure histories to predict the failure time of suspension histories may lead to erroneous estimates. In contrast, the proposed model only ensures that the output of suspension histories satisfies the constraints without

explicitly predicting their failure time. By using adversarial training, the model generalization for noise and multi-mode data is significantly improved. Regarding future research, transfer learning can be analyzed, especially for the RUL prediction under multiple failure modes when the equipment only has a few histories.

Chapter 3: RUL prediction of machines across different conditions using limited target-domain suspension histories

Historical data typically pertains to the operation of machinery under specific conditions, which can cause prognostic models trained on this data to struggle when applied to new environments. In practice, it may be feasible to collect limited suspension histories of the machinery in the new environment during planned maintenance. By utilizing these unlabeled data, transfer learning proves to be effective for cross-domain prognostics through domain adaptation. Therefore, this chapter focuses on developing transfer learning to examine how these limited suspension histories can be leveraged for machine prognostics across different conditions. The materials in this chapter are covered by the second research topic (Topic #2), which is introduced in Section 1.3. The organization of this chapter is as follows. In Section 3.1, an introduction to the reported studies on machine prognostics involving cross-domain conditions is made. In Section 3.2, the primary domain adaptation knowledge and the cross-domain RUL prediction problem are formulated. Section 3.3 describes the proposed transfer prognostic method. Section 3.4 validates the proposed methods using two case studies. Conclusions are made in Section 3.5. The results of this chapter are documented in a journal paper [176].

3.1 Introduction

A prognostic model based on deep learning is typically created offline and used to predict the RUL when new condition data is monitored. If a prognostic model is well constructed, it can be used online as long as all possible working conditions are included in the training dataset. However, traditional deep learning methods have a requirement that the measurements of training and testing samples follow a similar distribution. This can be a limitation because equipment often operates under different working conditions. The introduction of new conditions leads to significant differences in their measurement distributions. For example, if a prognostic model for a pump is built using failure histories collected at speeds ranging from 1000 rpm to 1200 rpm, it may not accurately predict failures when the pump operates at 1500 rpm. To overcome this limitation, the planned inspection can be employed to collect a small amount of target-domain history data when

the equipment operates in a new environment. This data, often known as “suspension histories”, is obtained when engineering assets are taken out of service before complete failure. By incorporating a few target-domain suspension histories, the prognostic model can improve its ability to generalize to the new environment.

As reviewed in Section 1.2.2, transfer learning has the capability to reduce differences in distributions by learning complex connections between two domains if we have access to data from the target domain. The typical approach to using transfer learning involves two steps. First, the “domain invariant feature” is extracted from the monitoring features using nonlinear models such as convolutional neural network (CNN) [64], kernel regression [65], or long short-term memory (LSTM) [66]. Then, these extracted features are aligned between the source and target domains through domain adaptation techniques by domain adaptation techniques such as minimizing their maximum mean discrepancy (MMD) [61]-[66] or using adversarial training [59][60]. In other words, a prognostic model based on transfer learning can leverage deep learning architectures to adapt to different domains and predict the RUL of a machine operating under new conditions.

In the initial operating stages, a machine is typically in a healthy state, and there are no noticeable differences in the measured data. However, after a certain period of time, known as the initial fault occurrence time (FOT), the machine begins to deteriorate. Many previous studies have overlooked the FOT or assumed a constant value for all equipment histories. However, as mentioned in [109], it is essential to identify FOT in order to estimate degradation levels for different machines. To address this, the hidden Markov model [63] and Pearson correlation coefficient analysis [61] have been developed to determine the FOT of machines with two-domain histories before adapting to a new domain. These methods assume that the entire life-cycle data is available in the target domain. As pointed out in [110], the RUL can be labeled on the target data when degradation states are recognized. However, in practice, the machine in the target-domain environment is often terminated in the early stages of degradation, making it challenging to label the RUL for these suspension histories due to the unknown failure time (FT) and bridge the gap between the two domains, especially when the RUL is close to zero. Only one previous study [68] has considered censored target-domain data in transfer prognostics by aligning the degradation direction.

Even though some progress has been made in cross-domain RUL predictions, there are still some shortcomings limiting the successful application of prognostic models in new environments. One major limitation is that existing work treats suspension histories in the target domain as unlabeled data without attempting to label their RUL value before domain adaptation. This means that supervised learning will not consider suspension histories, resulting in the inability to accurately predict RUL under multiple working conditions. Current domain adaptation methods only align degradation histories between two domains in a holistic manner. They assume that the conditional probability distributions from both domains are similar and only match the marginal probability distributions in domain adaptation. However, this assumption is not practical for solving the prognostic problem [111].

In this chapter, a transfer learning method is proposed for cross-domain prognostics using enough source-domain failure histories and limited target-domain suspension histories. The novel contributions of this chapter include: 1) FOT and FT are identified by developing COSMO-based health indicators. This allows us to label the RUL of all training data based on their health status, enabling the effective use of suspension histories. 2)A joint MMD method incorporated with manifold regularization is proposed. This method guides features from different domains to have consistent degradation trajectories. As a result, the source data can help prognostic models predict the RUL of target data. 3) A heuristic method and a parallel framework are proposed for model training and RUL predictions. These methods allow for the validation of model parameters and model uncertainties.

3.2 Preliminary and problem formulation

This study aims to investigate the RUL prediction problem in the cross-domain case. We assume that the source domain contains enough failure histories, but the target domain includes only limited suspension histories. The machine experiences multiple operating conditions in each domain, and the measurements in different domains have different distributions. Let $D^{(S)} = \{D_1^{(S)}, D_2^{(S)}, ..., D_m^{(S)}\}$ denote m failure histories in the source domain, where $D_i^{(S)}$ is the i-th failure history with $l_{f,i}$ inspection points, represented by $D_i^{(S)} = \{(x_{i,1}^{(S)}, t_{i,1}^{(S)}), (x_{i,2}^{(S)}, t_{i,2}^{(S)}), ..., (x_{i,lfi}^{(S)}, t_{i,lfi}^{(S)})\}$, and $(x_{i,j}^{(S)}, t_{i,j}^{(S)})$ is the j-th inspection point of the i-th failure history, where $x_{i,j}^{(S)} \in R^{1\times d}$ has d condition monitoring measurements (or features) and $t_{i,j}^{(S)}$ is the inspection time. Similarly, let

$D^{(T)} = \{D_{m+1}^{(T)}, D_{m+2}^{(T)}, ..., D_{m+n}^{(T)}\}$ represent n suspension histories in the target domain, and often $m \gg n$. Each suspension history has $l_{s,i}$ inspection points, which is denoted by $D_i^{(T)} = \{(x_{i,1}^{(T)}, t_{i,1}^{(T)}), (x_{i,2}^{(T)}, t_{i,2}^{(T)}), ..., (x_{i,lfi}^{(T)}, t_{i,lsi}^{(T)})\}$, where $(x_{i,j}^{(T)}, t_{i,j}^{(T)})$ is the j-th inspection point of the i-th suspension history, and the dimension and measurement types of $x_{i,j}^{(T)}$ and $t_{i,j}^{(T)}$ are the same as the source-domain histories. We call the last inspection point of the i-th failure history $(x_{i,lfi}^{(S)}, t_{i,lfi}^{(S)})$ the failure point, and also, $t_{i,lfi}^{(S)}$ can be regarded as the FT of the i-th failure history. In contrast, the last inspection time of the i-th suspension history $t_{i,lfi}^{(S)}$ is called the suspension time.

Oftentimes, a machine undergoes a stable period in the early stage, which means the machine is at the health stage. After FOT, the machine starts to deteriorate over time and is at the degradation stage. In this work, the output of the prognostic model, denoted by $\hat{y}$, is a value from 0 to 1, indicating the RUL percentage of the monitoring data. If the machine is at the health stage, its RUL percentage value will be 1. Otherwise, its RUL percentage is determined as a linear function related to the inspection time, FT, and FOT. As an example, the RUL percentage $y_{i,j}^{(S)}$ in terms of the inspection data $x_{i,j}^{(S)}$ will be:

$$y_{i,j}^{(S)} = \frac{t_{i,lfi}^{(S)} - t_{i,j}^{(S)}}{t_{i,lfi}^{(S)} - t_{i,fo}^{(S)}} \tag{3.1}$$

where $t_{i,fo}^{(S)}$ is the FOT of the i-th failure history in the source domain. Since $y_{i,j}^{(S)}$ is a percentage value from 0 to 1, the prediction of training data is not required to be standardized in the data preprocessing.

In this work, the cross-domain RUL prediction problem is formulated as: given enough failure histories in the source domain $D^{(S)}$ and limited suspension histories in the target domain $D^{(T)}$, construct a nonlinear prognostic model $\hat{y} = f(x)$, so that $f(x)$ can get good prognostic performance for the target-domain data.

In most machine learning methods, the input space of training and testing data should follow the same distribution. If a learned RUL prediction model can be well generalized to a new machine, the new machine should be the same as the machine in the training data and have the same working conditions, so that their measurements will have similar distributions. In practice, however, the

machine entity in the target domain often has new operating conditions compared to the machine in the source domain, resulting in significant differences in their measurement distributions.

Let y denote the RUL percentage and x represent the monitoring measurements or features. The measurements from the source and target domain are defined as $x^{(S)}$ and $x^{(T)}$, respectively. Since $y = Q(y \mid x)P(x)$, in order to use the prediction model for the target domain, it is required to ensure that $P(x^{(S)})$ is close to $P(x^{(T)})$ and $Q(y \mid x^{(S)})$ is also close to $Q(y \mid x^{(T)})$. Reported transfer prognostic works often assume that $Q(y \mid x^{(S)}) = Q(y \mid x^{(T)})$, and they only minimize the divergence of marginal probability distribution between the source and target domain. Fig. 3.1 illustrates a general framework to address the cross-domain RUL prediction problem. Since $P(x^{(S)}) \neq P(x^{(T)})$, we can assume that there is a feature mapping ϕ, so that the distribution of the mapped data $P(\phi(x^{(S)}))$ is equal to $P(\phi(x^{(T)}))$, where the $\phi(x)$ can be called the "domain invariant features". In work [57], the consensus self-organizing model (COSMO) is used to extract the domain invariant features from both domains, and more commonly in works [61][62][68][110], the neural network and the maximum mean discrepancy (MMD) are utilized respectively for the feature mapping and divergence construction.

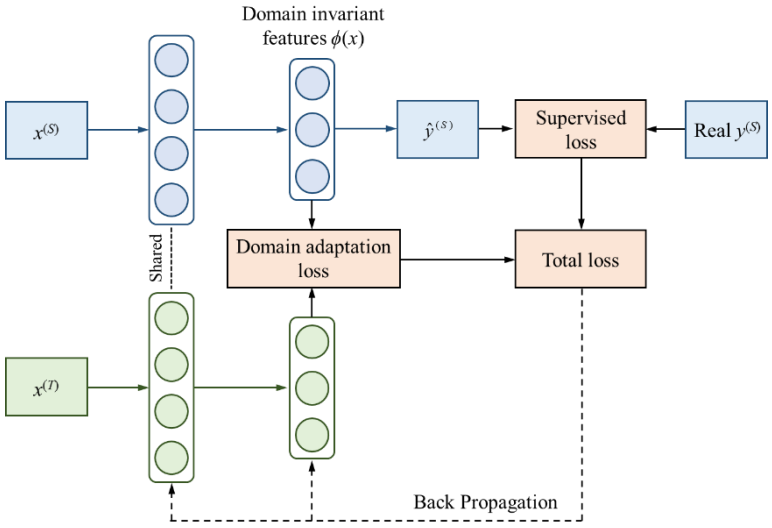


Fig. 3.1: Domain adaptation in prognostics

In most cases, the above methods cannot perform well on target-domain RUL predictions, since the assumption of $Q(y \mid x^{(S)}) = Q(y \mid x^{(T)})$ is often unsatisfied in prognostics. According to Bayesian theory $Q(y \mid x) \propto Q(x \mid y)P(y)$, if $Q(x^{(S)} \mid y) = Q(x^{(T)} \mid y)$, the condition $Q(y \mid x^{(S)}) = Q(y \mid x^{(T)})$ will be satisfied. However, if a pair of measurements $x^{(S)}$ and $x^{(T)}$ from two domains have the

same RUL, the distribution of $x^{(S)}$ and $x^{(T)}$ will also be significantly different. When we only consider minimizing the divergence of marginal probability distributions in two domains, the learned model may mistakenly believe that the distribution of all $x^{(T)}$ in the target domain is precisely the same, even considering a large amount of unlabeled target-domain data in model training. In other words, the predicted RUL for each $x^{(T)}$ may be almost identical without considering conditional probability distributions, and this result is quite common in related works.

3.3 Proposed methodology

3.3.1 Overview

If a prognostic model performs well in the target domain, its domain invariant features hold not only similar marginal distributions but also similar conditional distributions under the same RUL percentage, that is, $P(\phi(x^{(S)})) = P(\phi(x^{(T)}))$ and $Q(y \mid \phi(x^{(S)})) = Q(y \mid \phi(x^{(T)}))$. The most challenging part of transfer prognostics is to extract domain invariant features with minimum divergence in two conditional distributions. To address this problem, the target-domain histories also need to be labeled and considered in the prognostic model training. Fig. 3.2 summarizes the procedure of the proposed method. Two key steps and the corresponding ideas are presented as follows:

1) Training data preprocessing: The first step aims to construct a health indicator (HI) for training data in both source and target domains. The FOT of each history and the FT of suspension histories are estimated based on HI so that the RUL percentage of two-domain histories can be labeled by Eq. (3.1). HI is constructed based on the distance between the inspection point and its health reference group. The key idea of FOT identification is that the monitoring data at the health stage should have a stable and almost constant HI. When HI increases to a threshold, this time point can be regarded as the FT of suspension history.

2) Transfer model training: The second step learns domain invariant features based on the labeled two-domain samples. To minimize the divergence of conditional probability distributions in two domains, the pseudo-class labels are generated first for all target-domain suspended measurements $x^{(T)}$. After that, the measurement correlation in different class labels is improved by minimizing distribution differences in each intra-class. Even though the suspension histories are limited, and the predicted FT of suspension histories

may not be true, if the measurements have a clear degradation manifold, the iterative training will make the model get better pseudo-RUL predictions and pseudo-class labels. As a result, the RUL of target-domain measurements can be predicted through a secondary prediction.

Once the domain invariant features are learned, the corresponding prognostic model can be used for RUL prediction when new target-domain condition data are observed. The details of the proposed method are introduced in the following subsections.

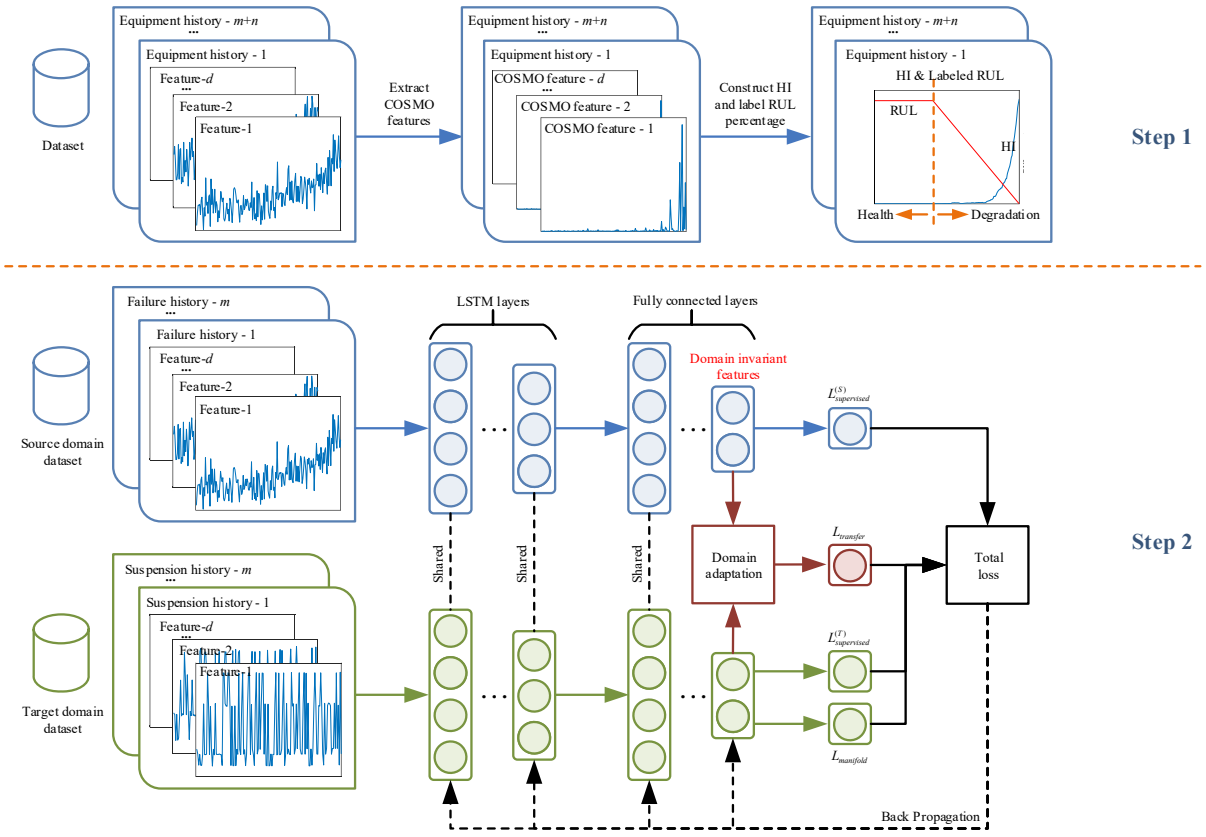


Fig. 3.2: Overview of the proposed transfer prognostic method

3.3.2 Data preprocessing

In this work, consensus self-organizing model (COSMO) features are generated first to evaluate machine health status. HIs of two-domain histories are constructed based on their COSMO features. Afterward, the FT and FOT adapted to each training history can be recognized, as illustrated in Step 1 of Fig. 3.2. COSMO was applied to detect deviations and faults in the previous

work [112]. As pointed out in the reported work [57], COSMO is especially effective for extracting degradation features from machines with multiple operating conditions.

The system deterioration is usually slight during the initial health stage and then aggravates over time during the degradation stage. Before estimating COSMO features, all monitoring measurements or features will be standardized into a standard normal distribution with zero mean and one variance. We define a small number of inspection points at the beginning of two-domain histories corresponding to relatively "healthy" systems and use these monitoring data as the health reference group by:

$$\begin{aligned} H_S &= \{x_{i,j}^{(S)} \mid 1 \le j \le \tau\},\ i = 1, 2, ..., m. \\ H_T &= \{x_{i,j}^{(T)} \mid 1 \le j \le \tau\},\ i = m+1, m+2, ..., m+n. \end{aligned} \tag{3.2}$$

where τ is the number of extracted inspection points. For example, if τ is set to 20, the first 20 inspection points of each equipment history are considered fairly "healthy".

The COSMO feature, denoted by $\theta_{i,j}^{(S)}$ and $\theta_{i,j}^{(T)}$ for source-domain and target-domain history, respectively, is a vector with the same dimension as the monitoring data. The COSMO features of the source and target domain measurements are computed respectively in the same approach based on their health reference group. Let $\theta_{i,j,p}$, $p = 1, \ldots, d$, denotes the p-th feature of $\theta_{i,j}$. The first step in computing $\theta_{i,j,p}$ is to measure the distance between the p-th feature of monitoring data $x_{i,j,p}$ and the p-th feature of all samples in the reference group H, so that we can get a set $\Omega_{i,j,p} \in R^{\tau \times 1}$ for each $x_{i,j,p}$. The absolute-value norms are commonly used to measure this distance as:

$$Dis\left(x_{i,j,p}, x_{H,q,p}\right) = \left|x_{i,j,p} - x_{H,q,p}\right|,\ with\ x_{H,q} \in H,\ q = 1, ..., \tau \tag{3.3}$$

where $x_{H,q,p}$ is the p-th feature of the q-th sample in the reference group H.

The COSMO feature $\theta_{i,j,p}$ is the minimum $\Omega_{i,j,p}$, denoting the minimum distance between the p-th feature of monitoring data and the p-th feature of all samples in the "healthy" group. To improve the model generalization, the K smallest values within $\Omega_{i,j,p}$ are selected based on the concept of the k-nearest neighbor, and $\theta_{i,j,p}$ is calculated as the mean of these smallest values by Eq. (3.4), where $\omega_{i,j,p}^{(k)}$ is the k-th smallest value in $\Omega_{i,j,p}$.

$$\theta_{i,j,p} = \frac{1}{K}\sum_{k=1}^{K}\omega_{i,j,p}^{(k)} \tag{3.4}$$

Based on the calculated COSMO features, the HI of $x_{i,j}$ can be modeled as Eq. (3.5). The constructed HI can possess an overall degradation trend. However, these HIs have local oscillations and cannot be used for estimating FOT and FT values. Therefore, a modified Weibull failure rate function [113] is utilized for HI fitting. This function is defined in Eq. (3.6), where β is the shape parameter, and η is the scale parameter. Parameter Y is introduced to scale the fitted HI to any range, and c denotes the initial HI when the time is 0. The nonlinear least square method is adopted to fit the HI functions. Double regression weights are given to the last 20% points in each history.

$$HI_{i,j} = \sum_{p=1}^{d}\theta_{i,j,p} \tag{3.5}$$

$$\lambda(t,\beta,\eta,c,Y) = c + Y\frac{\beta}{\eta^{\beta}}t^{\beta-1} \tag{3.6}$$

FOT and FT can be defined based on the fitted HI and preset thresholds. Since different equipment individuals have different initial HI values, only the dynamic degradation part, i.e., $Y(\beta/\eta^{\beta})t^{\beta-1}$, is considered, so that the fitted machine HI is always zero at the beginning and larger than zero during life but is not bounded at the end of life. Fig. 3.3 shows an example for estimating FOT and FT. First, a threshold value, denoted by Th_{fot}, is set to a minimal value for FOT identification. The healthy state is represented by the line with HI equal to Th_{fot}, since the degradation process of equipment is relatively slow during this interval. Considering different individuals have different degradation endpoints, the failure thresholds, denoted by $Th_{ft,i}$, $i = m+1, m+2,\ldots, m+n$, are set to values related to the fitted HI $HI_{i,lsi}$ at the suspended time. Simply, this work set $Th_{ft,i}$ to 1.5 times the $HI_{i,lsi}$ value, which is used only to give an initial orientation to the model training. With the determined FT and FOT, the RUL percentage of two-domain histories can be labeled by Eq. (3.1), and the labeled source-domain data $D^{(S)} = \{(x_{i,j}^{(S)}, y_{i,j}^{(S)})_{j=1}^{l_{fi}}\}_{i=1}^{m}$ and target-domain data $D^{(T)} = \{(x_{i,j}^{(T)}, y_{i,j}^{(T)})_{j=1}^{l_{si}}\}_{i=m+1}^{m+n}$ can be used in model training.

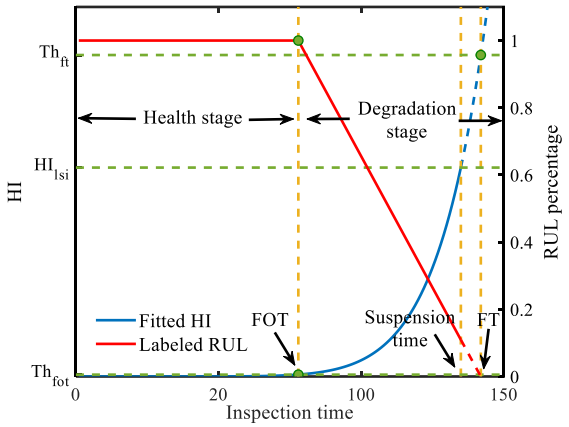


Fig. 3.3: Determination of FOT and FT based on fitted HI

3.3.3 Construction of transferable neural network

This work provides a general domain adaptation method for transfer prognostics by minimizing the distance between marginal and conditional probability distributions in two domains. The proposed transfer learning structure as shown in Step 2 of Fig. 3.2 is formulated as:

$$f = \arg\min_{f} L_{supervised} + \lambda_t L_{transfer} + \lambda_m L_{manifold} \tag{3.7}$$

where f is an NN-based regression model, $L_{supervised}$ is the supervised loss used for the supervised training, $L_{transfer}$ is the domain adaptation loss used to minimize the distance between the marginal and conditional probability distributions in two domains, $L_{manifold}$ is the manifold regularization loss used to guide the domain invariant feature to have a clear and continuous manifold, and λ_t and λ_m are weighting parameters. Due to the capability to cope with the regression of time series, LSTM is selected as the basic model f in this work. No layers are frozen during the model training, since deep learning networks without freezing can often get better transform performance [114].

The supervised prediction loss $L_{supervised}$ aims to learn a good mapping relationship between measurements and RUL based on the labeled data. This mapping relationship can guide the model to learn the degradation trace of two-domain data to get satisfactory pseudo-class predictions for the target-domain history. The mean squared error (MSE) is used to define the supervised loss as:

$$L_{supervised} = \mathbb{E}_{x^{(S)},y^{(S)}\sim p_{data}(x^{(S)},y^{(S)})}\left[f(x^{(S)}) - y^{(S)}\right]^2 + \lambda_t \mathbb{E}_{x^{(T)},y^{(T)}\sim p_{data}(x^{(T)},y^{(T)})}\left[f(x^{(T)}) - y^{(T)}\right]^2$$
$$= \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{l_{f,i}}\frac{1}{l_{f,i}}\left[f(x_{i,j}^{(S)}) - y_{i,j}^{(S)}\right]^2 + \lambda_t \frac{1}{n}\sum_{i=m+1}^{m+n}\sum_{j=1}^{l_{s,i}}\frac{1}{l_{s,i}}\left[f(x_{i,j}^{(T)}) - y_{i,j}^{(T)}\right]^2 \quad (3.8)$$

where $p_{data}(x^{(S)}, y^{(S)})$ and $p_{data}(x^{(T)}, y^{(T)})$ are the joint probability distribution of measurements and RUL percentage in $D^{(S)}$ and $D^{(T)}$, respectively, and λ_t is the overall weight for target-domain samples. The target-domain supervised loss should have a larger weight than the source-domain loss because this prediction model will eventually be used for the target domain. Different types of supervised loss functions can also be used in the proposed approach, such as mean absolute error and root mean squared error (RMSE).

Domain adaptation is performed to learn domain invariant features between the source domain and the target domain. In this work, the distance between the marginal and conditional probability distributions in two domains are minimized simultaneously. According to the Bayesian theory, we minimize the difference of conditional probability distributions $Q(y \mid x^{(S)})$ and $Q(y \mid x^{(T)})$ indirectly by minimizing the divergence between $Q(x^{(S)} \mid y)$ and $Q(x^{(T)} \mid y)$. Therefore, the domain adaptation loss is defined as:

$$L_{transfer} = \lambda_P d\left(P\left(\phi(x^{(S)})\right), P\left(\phi(x^{(T)})\right)\right) + \lambda_Q d\left(Q\left(\phi(x^{(S)}) \mid y\right), Q\left(\phi(x^{(T)}) \mid y\right)\right) \quad (3.9)$$

where $d(\cdot)$ represents the distance between the distributions in two domains, and λ_P and λ_Q denote the weights of the two tasks. An adaptive approach proposed in [115] can be used to set appropriate weights for these two distances. This work is dedicated to studying a general case with the same weights, that is, $\lambda_P = \lambda_Q$.

Similar to the reported works mentioned in Section 3.2.2, MMD is used to measure domain divergence. MMD quantifies the divergence as the squared distance between the instances in the reproducing kernel Hilbert space (RKHS). The distance of marginal probability distribution between the source domain and the target domain can be defined as Eq. (3.10). In this formulation, $\mathcal{H}$ denotes the RKHS. Following the previous works [116], a mixture of five Gaussian kernels is used since the multiple kernels in MMD often lead to better performance.

$$d\left(P\left(\phi(x^{(S)})\right),P\left(\phi(x^{(T)})\right)\right)=\left\|\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{l_{f,i}}\phi(x_{i,j}^{(S)})-\frac{1}{n}\sum_{i=m+1}^{m+n}\sum_{j=1}^{l_{s,i}}\phi(x_{i,j}^{(T)})\right\|_{\mathcal{H}}^{2} \tag{3.10}$$

Furthermore, the entire life span data in the source domain will be divided into C classes based on their RUL percentage, denoted by *Class* c, c = 1, 2, …, C. The classification depends on the distribution of data in the source domain. If there is a significant difference in the distribution of measurements between two RUL intervals, it can be divided into two classes. In this work, the failure history is classified into eight classes. The first class contains the inspection points with a RUL percentage of 1. The latter three classes are divided at every 0.2 interval of RUL percentage. As the machine is closer to failure with the decrease of RUL, the life span data is divided at a smaller 0.1 interval from class C_5 onward. With this setting, the source-domain data will be divided into multiple classes, represented by $D_c^{(S)}$.

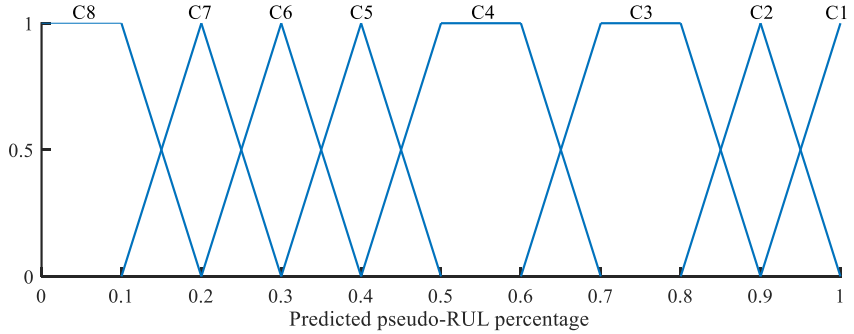


Fig. 3.4: Fuzzy set for labeling target-domain histories

Let $D_c^{(T)}$ denote the target domain data with the c-th pseudo-class label. The distance between the conditional probability distributions in the source and target domains can then be constructed as the sum of the MMD between the class means in two domains by Eq. (3.11), where n_c and m_c are the number of samples in $D_c^{(S)}$ and $D_c^{(T)}$, respectively.

$$d\left(Q\left(\phi(x^{(S)})\,|\,y\right),Q\left(\phi(x^{(T)})\,|\,y\right)\right)=\sum_{c=1}^{C}\left\|\frac{1}{n_c}\sum_{x^{(S)}\in D_c^{(S)}}\phi(x^{(S)})-\frac{1}{m_c}\sum_{x^{(T)}\in D_c^{(T)}}\phi(x^{(T)})\right\|_{\mathcal{H}}^{2} \tag{3.11}$$

To estimate the pseudo-class label for the suspension histories in the target domain, the fuzzy set [117] as shown in Fig. 3.4 is used to transform the regression problem into a classification problem similar to the reported work [118]. As an example, suppose that the predicted pseudo-RUL

percentage of one target-domain sample is 0.35 at an iterative epoch, then this sample will be classified into Class 5 or Class 6 with the same probability.

Even though both marginal and conditional probability distributions in the two domains are matched, the method still cannot perform a fully valid RUL prediction on the target-domain data. The reason is that all suspension histories in the target domain are terminated before complete failures. Domain adaptation with complete source data and partial target data may result in mismatches at the degradation level close to the failure. Therefore, manifold regularization is adopted to guide the learned domain invariant feature to have a clear and continuous manifold, so that the prognostic model can predict the RUL of target data close to the failure. The manifold regularization loss modified by [119] is defined as Eq. (3.12) with $L_i = I_i - D_i^{-1/2} W_i D_i^{1/2}$. In this formulation, L_i is the normalized graph Laplacian matrix for the i-th suspension history in the target domain, I_i is the identity matrix with a dimension of $l_{s,i} \times l_{s,i}$, and D_i is a diagonal matrix for L_i given by $D_{i,j_1,j_1} = \sum_{j_2=1}^{l_{s,i}} W_{i,j_1,j_2}$. W_i is the data adjacency graph calculated based on HI constructed in Section 3.3.2 by Eq. (3.13), where $\mathcal{N}(x_{i,j}^{(T)})$ is the set composed of the nearest neighbors of $x_{i,j}^{(T)}$.

$$\begin{aligned} L_{manifold} &= \sum_{i=m+1}^{m+n} \sum_{j_1,j_2=1}^{l_{s,i}} \left(\phi\left(x_{i,j_1}^{(T)}\right) - \phi\left(x_{i,j_2}^{(T)}\right)\right)^T \left(\phi\left(x_{i,j_1}^{(T)}\right) - \phi\left(x_{i,j_2}^{(T)}\right)\right) W_{i,j_1,j_2} \\ &= \sum_{i=m+1}^{m+n} \sum_{j_1,j_2=1}^{l_{s,i}} \phi\left(x_{i,j_1}^{(T)}\right)^T L_{i,j_1,j_2} \phi\left(x_{i,j_2}^{(T)}\right) \end{aligned} \tag{3.12}$$

$$W_{i,j_1,j_2} = \begin{cases} \left|HI_{i,j_1} - HI_{i,j_2}\right|, & if\ x_{i,j_2}^{(T)} \in \mathcal{N}(x_{i,j_1}^{(T)}) \\ 0, & others \end{cases} \tag{3.13}$$

We can directly determine the neighbors $x_{i,j}^{(T)}$ according to their inspection times. For the j-th point of the i-th suspension history $x_{i,j}^{(T)}$, as an example, the set of neighbors can be defined as $\{x_{i,j-1}^{(T)}, x_{i,j+1}^{(T)}\}$, so that the corresponding data adjacency graph W_i is constructed by Eq. (3.14). Minimizing $L_{manifold}$ can guide the degradation trajectory to be properly projected into a new subspace in a continuous low-dimensional form. The intrinsic geometry of the loss function will depend on the manifold of the monitoring measurements. Therefore, the mapped degradation path will be relatively flat in the healthy stage, while the slope of the RUL degradation will increase in the degradation stage.

$$W_i = \begin{pmatrix} w_1 & & 0 \\ & \ddots & \\ 0 & & w_{lsi} \end{pmatrix}, \text{ with } w_j = \begin{pmatrix} W_{i,j,j-1} & 0 & W_{i,j,j+1} \end{pmatrix} \tag{3.14}$$

3.3.4 Model training and cross-domain prognostics

Multiple tasks are designed in the proposed model based on the domain adaptation strategy. Given the loss functions listed above, the total objective function can be formulated as:

$$L = \lambda_1 \left(L_{supervised}^{(S)} + \lambda_2 L_{supervised}^{(T)} \right) + L_{transfer} + \lambda_3 L_{manifold} \tag{3.15}$$

where $L_{supervised}^{(S)}$ and $L_{supervised}^{(T)}$ are the supervised loss in Eq. (3.8) for the source and target data, respectively, and λ_1, λ_2, and λ_3 denote the weights of multiple tasks in transfer learning. The weight of each loss determines the importance of the corresponding optimization task. Since $L_{supervised}^{(T)}$ should have a larger weight than $L_{supervised}^{(S)}$, we set λ_2 to a constant value of five in this work. The manifold regularization task can provide additional evidence for the relevance or irrelevance of domain invariant features. As a result, $L_{manifold}$ can be regarded as a regularizer for domain adaptation. The weight λ_3 will not strongly determine the model performance, so we set λ_3 to one throughout the case studies. Weight λ_1 is a positive adjustable parameter, which is essential to the performance of the transfer learning algorithm.

When we have available labeled data in the target domain, cross-validation can be used to determine the optimal weights and model structure. In the formulated problem, however, no data is labeled in the target domain. Therefore, we follow a heuristic idea to determine whether the setting of λ_1 is reasonable. Let d_p and d_q denote the distance of the marginal probability distribution, i.e., Eq. (3.10), and the conditional probability distribution, i.e., Eq. (3.11), in two domains, respectively. During the model training, d_p and $L_{manifold}$ will always decrease steadily. If the setting of λ_1 is reasonable, $L_{supervised}^{(S)}$ and $L_{supervised}^{(T)}$ will also decrease steadily in each iteration, but d_q will first rise sharply, then fall, and finally converge. That is because in the early training iterations, the predicted pseudo-class labels of target-domain histories are almost the same, so both d_p and d_q are small and in the same magnitude order. As the supervised loss decreases, the prognostic model can get better and better pseudo-class predictions. Due to the increase in the number of target data with different pseudo-class labels, the distance between the conditional probability distributions

in two domains will increase. When the supervised loss converges, d_q will gradually decrease and converge, indicating that the domain adaptive training is completed. Therefore, if d_q consistently decreases and eventually converges during training, the weight of supervised loss λ_1 needs to be further improved.

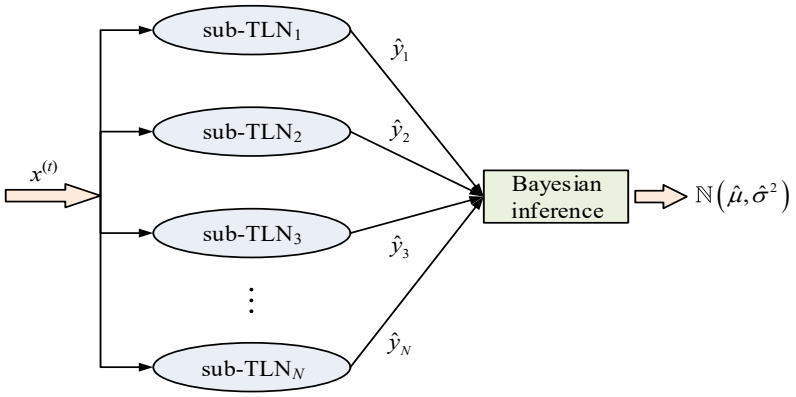


Fig. 3.5: Comprehensive framework of the proposed prognostic model

Due to the lack of labeled data in the target domain, it is impossible to verify which configured model is optimal. Therefore, we train different configurations in a parallel fashion to consider model uncertainties, as shown in Fig. 3.5. The proposed prognostic model for RUL predictions consists of multiple parallel-connected transfer learning networks, denoted by *TLN*. Let *N* transfer learning subnets denote as sub-TLN_{nt}, nt = 1, 2,…, *N*. After completing the model training of *N* sub-*TLN* based on a preset maximum epoch number, we can get *N* different predicted RUL values, denoted as $\hat{y}_{nt}$, nt = 1, 2,…, *N*, when a set of condition data is observed. We assume that the predicted RUL percentage follows a Gaussian distribution with an unknown mean $\hat{\mu}$ and variance $\hat{\sigma}^2$. According to the Bayesian inference [120], a Markov chain can be generated by randomly sampling μ and σ^2 in sequence, and the $\hat{\mu}$ and $\hat{\sigma}^2$ can be estimated as the mean of the generated values by:

$$\sigma_{mc,i}^2 \sim \Gamma^{-1}\left(\frac{N\text{-}1}{2}, \frac{1}{2}\sum_{nt=1}^{N}\left(\hat{y}_{nt}\text{-}\bar{\hat{y}}\right)^2\right), \mu_{mc,i} \sim N\left(\frac{1}{N}\sum_{nt=1}^{N}\hat{y}_{nt}, \frac{\sigma_{mc,i}^2}{N}\right) \tag{3.16}$$

$$\hat{\sigma}^2 = \frac{1}{l_{mc}}\sum_{i}^{l_{mc}}\hat{\sigma}_{mc,i}^2, \hat{\mu} = \frac{1}{l_{mc}}\sum_{i}^{l_{mc}}\hat{\mu}_{mc,i} \tag{3.17}$$

where $\mu_{mc,i}$ and $\sigma^2_{mc,i}$ are the *i*-th sampled mean and variance, $\bar{\hat{y}}$ is the mean of the predicted RUL values $\hat{y}_{nt}$, and l_{mc} is the length of the generated Markov chain. It is noted that even though the Gaussian distribution assumption is not verified in this study, it can be established through the use of normality tests on the results of various models. When the number of constructed sub-TLN models is small, Bayesian inference can provide more effective estimates than traditional methods such as maximum likelihood estimation (MLE).

3.4 Case studies

In the case study, two benchmark datasets are considered. In Section 3.4.1, the comparative methods and the performance metrics are introduced. The C-MAPSS dataset and XJTU-SY bearing dataset are used respectively to validate the cross-domain prognostic performance of the proposed method in Section 3.4.2 and Section 3.4.3. The results are discussed in each section.

3.4.1 Comparative methods and performance metrics

In order to demonstrate the superiority of the proposed transfer prognostic method, five models are used for comparison, which are LSTM-S, LSTM-T, NN-based transfer component analysis (TCA), domain adversarial neural network (DANN), and transfer multi-layer perceptron (TMP). LSTM-S and LSTM-T are baseline methods that follow the typical supervised learning paradigm without domain adaptation. LSTM-S uses source data for model training, while LSTM-T uses target-domain suspension histories with real RUL labels for model training. TCA is a transfer learning method proposed in [61], which utilizes NN and MMD to map the features and minimize the distribution divergence. DANN proposed in [59] uses a classifier to measure the distribution divergence and learn the domain invariant features by adversarial training. TMP proposed in [63] combines the above two works and uses MMD and a classifier for the domain adaptation. We construct LSTM with the same structure for each method and consider domain adaptation only on the last layer, i.e., domain invariant features. Reported transfer learning-based methods only treat target-domain data as unlabeled data without attempting to label RUL for suspension histories.

To evaluate the accuracy of the proposed method, the root mean squared error (RMSE) is evaluated as:

$$RMSE = \sqrt{\frac{1}{n_t}\sum_{i=1}^{n_t}\left(RUL_{ture} - RUL_{predicted}\right)^2} \tag{3.18}$$

where n_t is the total number of the test data in the target domain, and RUL_{true} and $RUL_{predicted}$ are the real and predicted RUL, respectively. We assume that the FOT of all test data is known so that the $RUL_{predicted}$ can be computed based on the estimated RUL percentage.

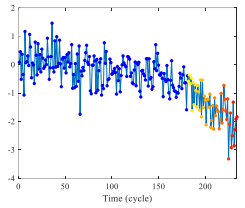
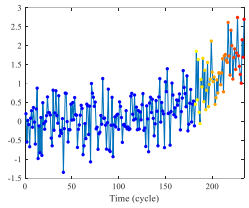
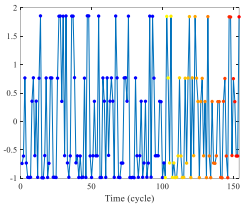
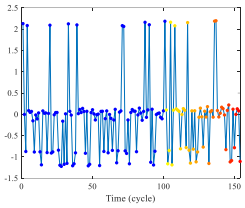
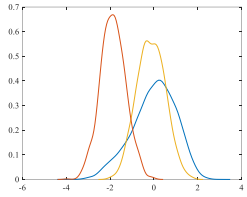
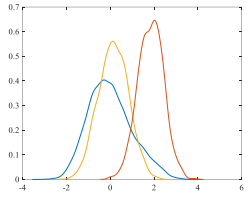
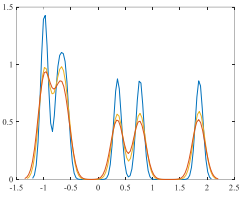
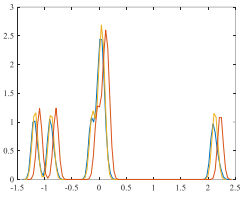
We repeat comparative methods ten times using the same LSTM structure with different random seeds. In contrast, the proposed method will be repeated ten times with different LSTM structures and model parameters, where each model is regarded as a sub-TLN. Model training is terminated according to the preset maximum epoch, and the final test RMSE is recorded for comparisons.

3.4.2 Case study 1: C-MAPSS dataset

In the first case study, the C-MAPSS dataset is adopted in a manner similar to Section 2.5.2, which was created by NASA to simulate the actual degradation of turbofan engines [106]. The C-MAPSS dataset includes four subsets, denoted by FD001, FD002, FD003, and FD004. Subset FD001 and FD003 are subjected to a single operating condition, while subset FD002 and FD004 are more complex due to six operating conditions. In this case, FD001 is considered as the source-domain data, and FD002 is used as the target-domain data. Fourteen sensors, i.e., sensors 2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20, and 21, are selected as the input measurements, and are normalized respectively in each domain. Examples of sensor readings, including Sensor 7 and 15, from FD001 and FD002, are shown in Table 3.1, where in the last row, the blue line, yellow line, and red line represent the sensor distributions with all life span, RUL percentage from 0.4 to 0.6, and RUL percentage from 0 to 0.1, respectively. Between FD001 and FD002, both the marginal and conditional probability distributions have significant differences.

In this work, 30 failure histories, i.e., the first 30 training engines in FD001, and only four suspension histories, i.e., the first four training engines in FD002, are used in the model training. The size of the time window is set to 10. The last 100 training engines in the FD002 subset are utilized as testing engines to validate the model performance. The suspension time is randomly selected from 0 to 30 cycles in advance of the failure time. Before the prognostic model training, all data are labeled with RUL percentage using the method proposed in Section 3.3.2. In this case, Th_{fot} is set to 1×10^{-4} for all source-domain data and set to 0.01 for all target-domain data, which

Table 3.1: Difference of two normalized sensor values between FD001 and FD002

Subsets	FD001		FD002	
	Sensor 7	Sensor 15	Sensor 7	Sensor 15
Trajectory #20 from training set				
Distribution divergence				

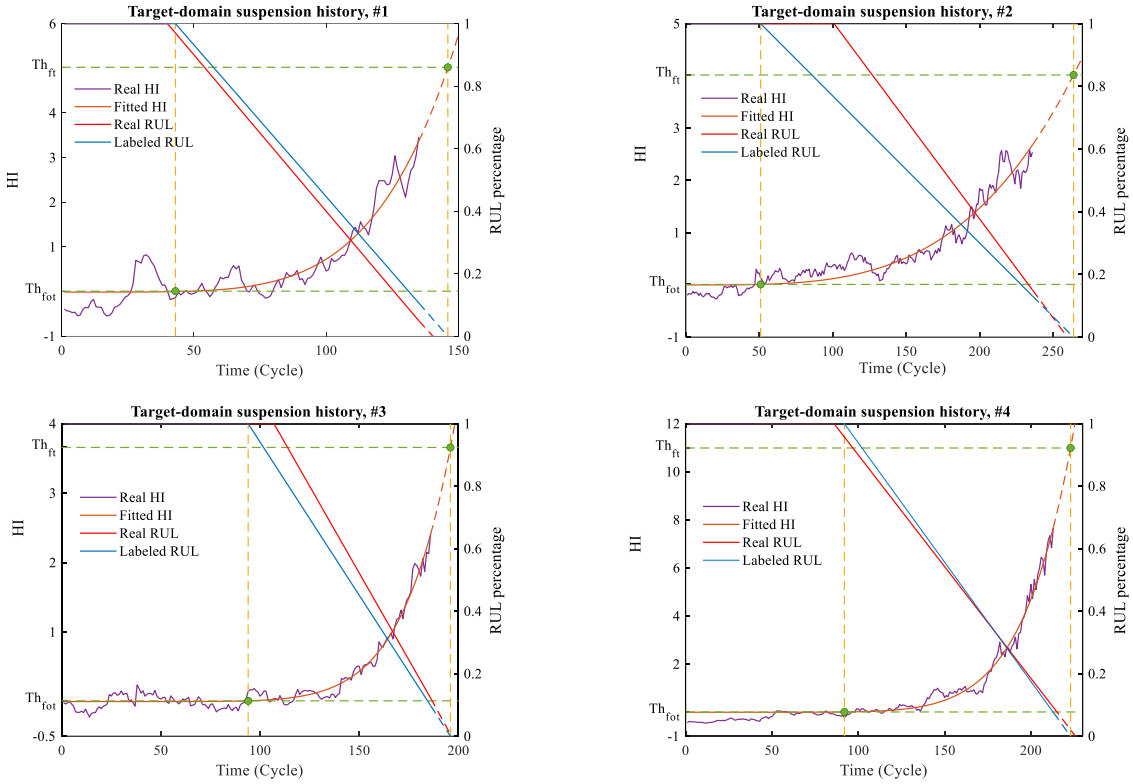


Fig. 3.6: Identification of FOT and FT of four target histories

Table 3.2: Configurations of ten sub models

Model	LSTM layers, (units)	Fully connected layers, (units)	Loss weights λ_1	Final test RMSE
Model 1	2, (64, 32)	1, (16)	300	21.6710
Model 2	2, (64, 64)	2, (32, 16)	500	22.6539
Model 3	2, (32, 32)	1, (16)	200	22.4510
Model 4	3, (64, 32, 16)	1, (16)	300	22.3893
Model 5	2, (64, 32)	1, (32)	500	21.8434
Model 6	3, (64, 64, 32)	1, (32)	200	21.9997
Model 7	2, (32, 16)	2, (32, 16)	300	21.8382
Model 8	2, (64, 32)	2, (16, 16)	500	22.1359
Model 9	3, (64, 32, 32)	1, (16)	200	22.5796
Model 10	2, (64, 32)	2, (32, 16)	300	21.9443

depends on the size of equipment histories. Fig. 3.6 presents the identified FOT and FT of four target-domain histories. Since the labeled RUL percentage is close to the real RUL percentage, the effectivity of the constructed HI can be validated. Table 3.2 lists ten model configurations and their final RMSE on the test sets. Ten different configurations are built in a parallel fashion since no labeled data is available for cross-validation. We can observe that not all deeper structures have better RMSEs on testing engines. Over-parameterized LSTMs will not directly cause significant overfitting, but conversely, partially over-parameterized structures have better generalization.

We use the same LSTM structure for the five comparative models, which have two LSTM hidden layers with 64 and 32 neurons, and one fully connected layer with 16 neurons, and the loss weights λ_1 is set to 300 for all models. Fig. 3.7 presents the comparative results of RMSE using four target domain histories, in which the middle line of each box denotes the mean value of 10 tests; the dark zone denotes the 95% confidence interval of the mean value; the shallow zone represents the region that contains about two-thirds of the data points [108]. Overview, the proposed method outperforms the other four methods because it has the lowest RMSE and shows a statistical difference between the RMSE produced by other models. When the training target data is limited, the supervised learning methods will cause significant overfitting, and therefore, the proposed method can still have a better performance than LSTM-T. Reported three transfer learning methods, i.e., TCA, DANN, and TMP, treat target-domain data as unlabeled data and only minimize the distance between the marginal probability distributions in two domains. It is

observed that the RMSEs of reported transfer learning methods are even higher than that of LSTM-S, that is, blindly using transfer learning may lead to counterproductive results.

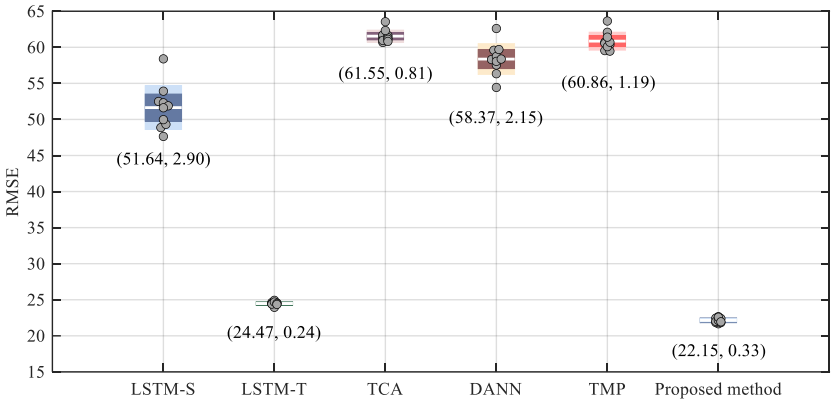


Fig. 3.7: RMSE of comparative methods with 4 target domain histories

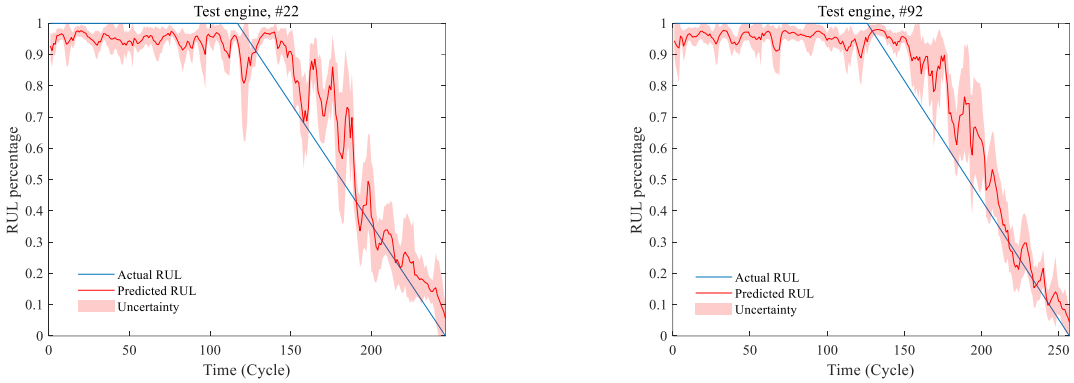


Fig. 3.8: RUL predictions for two testing engines using the proposed method

Fig. 3.8 shows the RUL predictions for two testing engines using the proposed method. Based on the Bayesian inference method in Section 3.3.4, 100 mean and variance values are generated by Eq. (3.16) so that the mean shown by the red line and variance of the distribution can be estimated. The uncertainty zone shown by the light red region ranges from μ-2σ to μ+2σ, which means there will be more than 95% predicted values in the uncertainty zone. As a result, the predicted RUL percentage is a Normal distribution $N(\mu, \sigma^2)$ rather than a single value. Since the training histories from testing engines are censored and very limited, there is no available near-failure target-domain data to guide model training. The manifold regularization can help the model track the degradation trend of testing engines, but the predicted RUL obtained by the proposed method may still have a small deviation at the end of the engine life. To obtain an exact failure time, the predicted RUL

values may be further fitted as suggested in [52]. Nevertheless, the proposed method can provide good prognostic performance for testing engines as the predicted RUL is close to the actual RUL during the engine degradation.

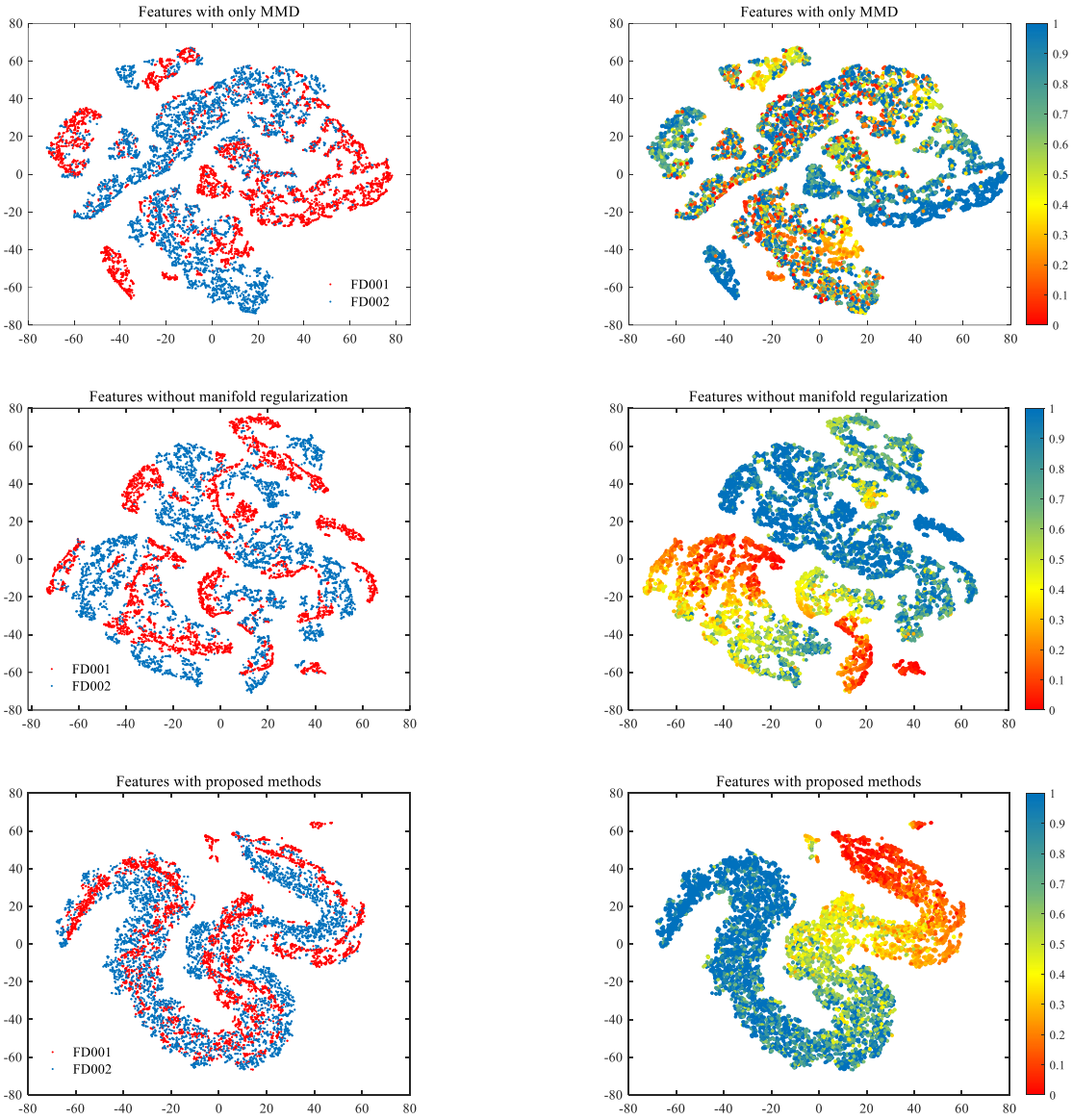


Fig. 3.9: Visualization of domain invariant features by different methods

Fig. 3.9 presents the manifold visualization of domain invariant features by different methods based on the t-distributed stochastic neighbor embedding (t-SNE), in which each subset contains 30 engine data. In the t-SNE graph, the x-axis and y-axis represent the two features after the dimensionality reduction process. If the prognostic model can work for the target domain, the

domain invariant features should have similar marginal and conditional probability distributions in the two domains. The manifold of the two domains should almost overlap in the left-column figure, and the two-domain data at similar positions should have a similar RUL percentage, as shown by the color in the right-column figure. When we only minimize the divergence of marginal probability distributions between two domains, the RUL percentage, i.e., the color of the conditional features, is chaotic and disordered, as shown in the first two figures in Fig. 3.9. Thus, the conditional probability distributions of two domains still have significant divergence. When we attempt to label target-domain histories and use proposed supervised loss and domain adaptation loss to train a prognostic model, the divergence of both marginal and conditional probability distributions in two domains can be simultaneously minimized, as shown in the middle two figures in Fig. 3.9. However, this manifold cannot show a clear degradation trace, leading to large deviations in RUL predictions near the end of life, as well as sensitive predictions for different model configurations. With the use of manifold regularization, the domain invariant features of two-domain histories will present clear, continuous, and similar manifolds, as shown in the last two figures in Fig. 3.9. As a result, the manifold of source-domain data can help the prognostic model predict the RUL of target-domain data so that the cross-domain prognostic performance can be further improved.

3.4.3 Case study 2: XJTU-SY bearing dataset

To further validate the proposed method, the bearing dataset provided by the Xi'an Jiaotong University and the Changxing Sumyoung Technology Company is adopted in the second case study [101]. As shown in Table 3.3, fifteen rolling element bearings are tested under three different operating conditions. The sampling frequency of the vibration signal is 25.6 kHz, and 32768 measurements are recorded in the interval of 1.28 s every 1 min. The bearing failure may be caused by inner race wear, outer race faults, and cage fracture. The accelerated degradation is stopped when the vibration amplitude goes over 10 times the maximum normal amplitude. Therefore, the bearings under different operating conditions will have different failure thresholds. In this case, two cross-domain RUL prediction tasks presented in Table 3.4 are investigated with different source-target pairs. The testing bearings have the same operating condition and similar failure behaviors as the training target-domain bearing.

Table 3.3: Information of the XJTU-SY dataset

Operating condition	Rotating speed (rpm)	Load (kN)	Bearing dataset
1	2100	12	Bearing 1_1, 1_2, 1_3, 1_4, 1_5
2	2250	11	Bearing 2_1, 2_2, 2_3, 2_4, 2_5
3	2400	10	Bearing 3_1, 3_2, 3_3, 3_4, 3_5

Table 3.4: Cross-domain prognostic tasks in the XJTU-SY bearing dataset

Task	Training data		Testing data
	Source domain	Target domain	
Task 1	Bearing 2_1, 2_2, 2_3, 2_4, 2_5	Bearing 1_1	Bearing 1_2, 1_3
Task 2	Bearing 2_1, 2_2, 2_3, 2_4, 2_5	Bearing 3_3	Bearing 3_1, 3_4

Table 3.5: Definition of RS features

RS features		Energy ratio features
Time-domain	Frequency-domain	Time-frequency-domain
F1: RS of 8 classical time-domain features, including RMS, kurtosis, peak-to-peak, crest factor, skewness, shape factor, clearance factor, and impulse factor.	F2: RS of [0,12,800] Hz F3: RS of [0,3200] Hz F4: RS of [3200,6400] Hz F5: RS of [6400,9600] Hz F6: RS of [9600,12800] Hz	F7: Energy ratio of (3,0) F8: Energy ratio of (3,1) F9: Energy ratio of (3,2) F10: Energy ratio of (3,3) F11: Energy ratio of (3,4) F12: Energy ratio of (3,5) F13: Energy ratio of (3,6) F14: Energy ratio of (3,7)

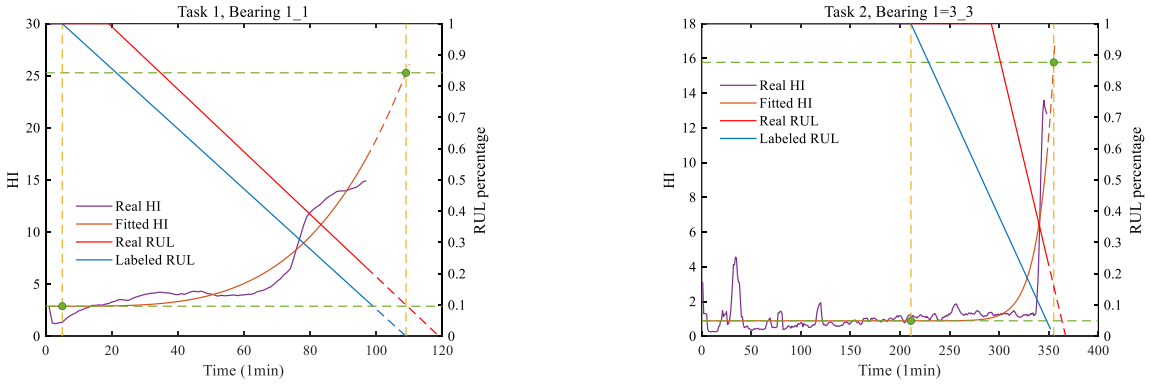


Fig. 3.10: Failure time prediction of training bearing data in the target domain

Table 3.6: Optional hyperparameters in the proposed method

Hyperparameter	Range
LSTM layers	{2, 3}
Units in LSTM layers	{8, 16, 32}
Full connected layers	{1, 2}
Units in full connected layers	{4, 8}
Loss weights λ_1	{1000, 1500, 2000}

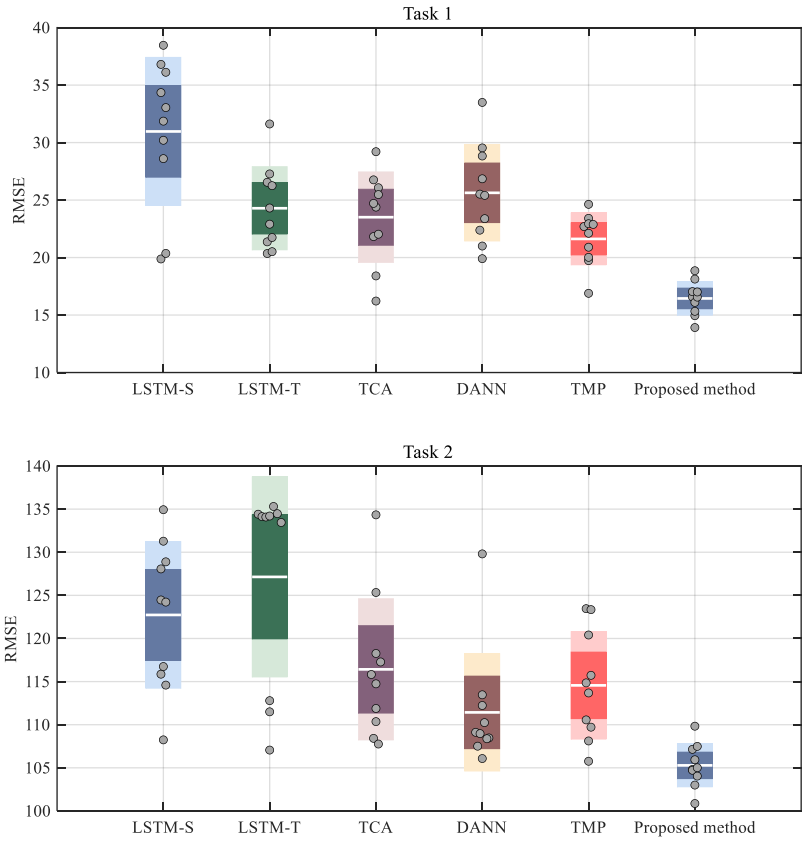


Fig. 3.11: RMSE of comparative methods using bearing dataset

Before modeling HI for RUL labeling, six related-similarity (RS) and eight energy ratio features referred to [52] are extracted from the vibration measurements in each direction. In this case, the size of the time window is set as 5. As shown in Table 3.5, eight classical time-domain features are integrated into one RS feature. A full frequency spectrum and four sub-band frequency spectra are used in the frequency domain to construct five RS features. In addition, the energy ratios of eight frequency sub-bands generated by a haar wavelet package transformed with three-level

decomposition are extracted as eight time-frequency features. All features are limited to a range from 0 to 1 for different bearings, and thus, these features can be used without normalization. Seven features are selected as the input features, including F1, F2, F3, F7, F9, F10, and F13, with the highest criteria value calculated by [52]. As a result, the dimension of input of the prognostic model is 14 since two-axis, i.e., horizontal and vertical, vibration signals are available. The training bearings in the target domain are truncated, and the suspension time is selected based on the principle that the vibration amplitude exceeds 25 g. Fig. 3.10 presents the labeled RUL of two target-domain bearing histories, in which the Th_{fot} is set to 1×10^{-4} to estimate the FOT of data.

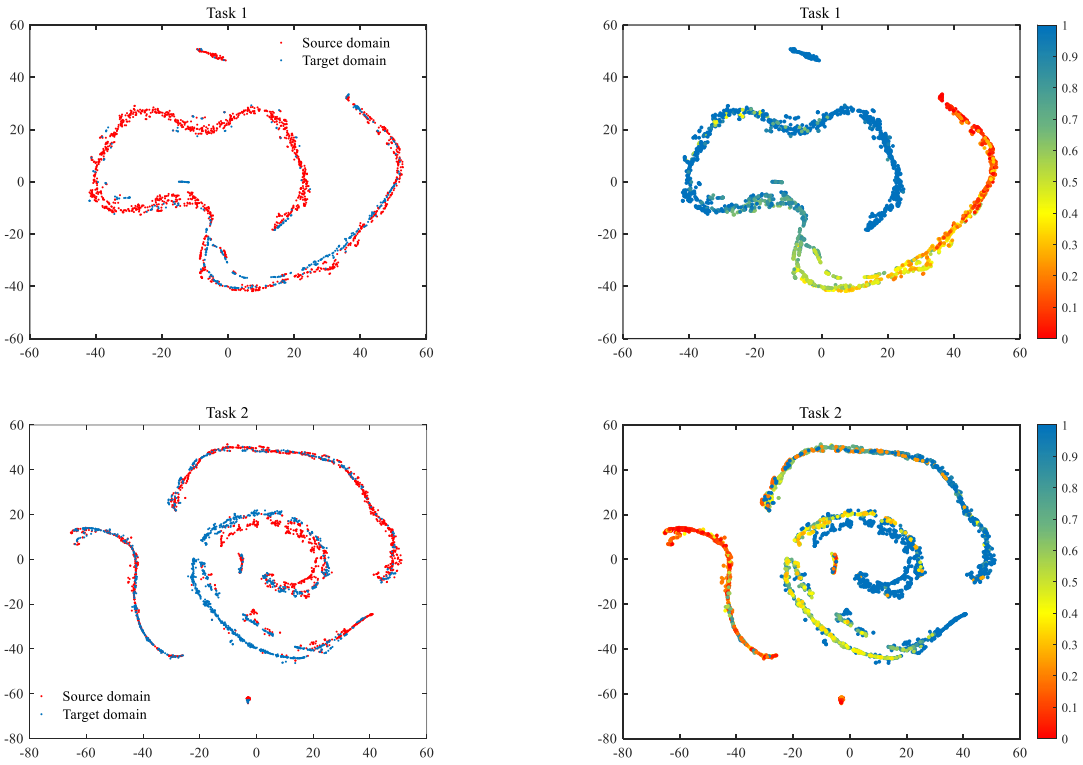


Fig. 3.12: Visualization of domain invariant features by the proposed method

Ten configurations are randomly selected from the optional hyperparameters listed in Table 3.6 for Task 1 and Task 2, respectively. The other four comparative methods use the same LSTM structure, which has two hidden layers with 32 and 16 neurons and one fully connected layer with 16 neurons, and the loss weights λ_1 is set to 2000 for all models. Fig. 3.11 presents the experimental results of RMSE using five methods. It can be observed that the proposed method outperforms the other four methods and shows a statistically significant difference in RMSE in both Task 1 and

Task 2. Some reported transfer learning methods can get better performance than supervised learning, such as LSTM-S and LSTM-T. Compared with case study 1, the available data in this case 2 is even scarcer. We only have five source-bearing data and one target-bearing data. As a result, the additional domain adaptation loss can be regarded as a regularizer for the supervised loss to reduce the risk of overfitting, thereby improving the generalization of the prognostic model.

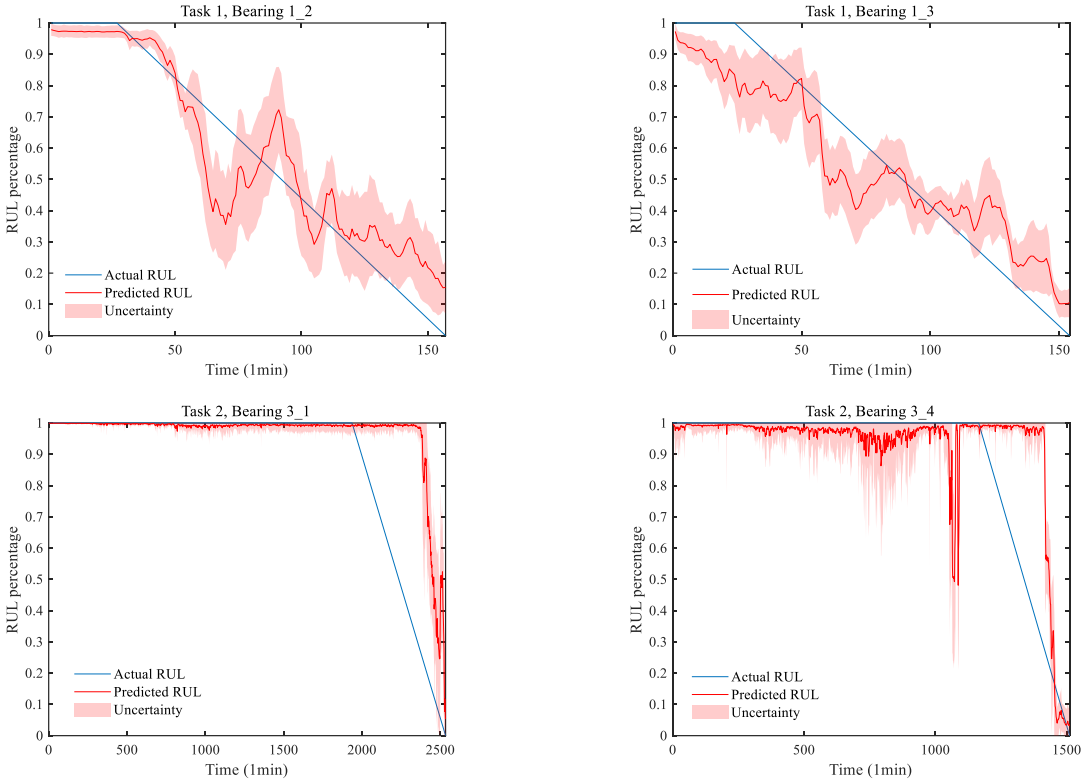


Fig. 3.13: RUL predictions for four testing bearings using the proposed method

The distributions of domain invariant features obtained by the proposed method are visualized as shown in Fig. 3.12. The target domain features are basically distributed on the manifold of the source domain features. In addition, the two-domain features at similar positions have a similar RUL percentage. As a result, the distance between the marginal and conditional probability distributions in two domains can be minimized simultaneously. In Task 2, the manifolds of two-domain histories are not completely continuous because the condition monitoring features of target-domain bearings are quite different between the health stage and the degradation stage. Nevertheless, since the manifolds of the two domains are almost matched in the marginal and conditional distributions, the prognostic model can be effectively used for the RUL prediction of

bearings in the target domain. Fig. 3.13 shows the RUL predictions for the testing bearings by using the proposed method. Overall, the predicted RUL is close to the actual RUL. Especially in the second task, the bearings will quickly degrade and completely fail after FOT, but the prognostic model can also find this abrupt fault and get good RUL predictions. The RUL predicted by the proposed method has lower uncertainties at the beginning of failure histories, possibly because there is more health-bearing data in the training dataset. In the second task, since the failure modes of Bearing 3_1 and Bearing 3_4 are significantly different from Bearing 3_3 used for training, the prognostic model of Bearing 3_1 and Bearing 3_4 undertaken with the proposed method has lower accuracy than other case studies, but still have significantly higher performance as compared to other reported methods as shown in Fig. 3.11. This case study illustrates the capability of the proposed method to achieve reliable cross-domain prognostics for the bearing dataset.

3.5 Conclusions

This chapter proposes a transfer learning framework for cross-domain prognostic tasks. To address the limitations of reported methods, we take advantage of COSMO to estimate FOT and FT for two-domain training histories. Afterward, a novel domain adaptation strategy is presented for transfer RUL predictions by minimizing the distance of both marginal and conditional probability distributions in different domains. Even though the procedure of estimating FOT and FT may not be perfect, the iterative domain adaptation training can make the prognostic model learn better and better domain invariant features in two domains. Domain invariant features will guide two-domain histories to have similar degradation manifolds so that the source data can help a small number of target data to improve the model generalization. The main conclusions from the case study validate that the proposed transfer prognostic method can improve the RUL prediction accuracy by more than 30% compared with different reported methods.

This study fills several gaps in traditional transfer prognostic methods. Different from reported transfer prognostics that only align two-domain histories in a holistic manner, this work is the first to simultaneously minimize the divergence between marginal and conditional probability distributions in two domains for transfer prognostics. In addition, manifold regularization is adopted for the first time to guide domain invariant features to have a clear and continuous manifold. Finally, since there is no available data in the target domain to support the model cross-

validation, a heuristic method and a parallel framework are proposed for model training and RUL predictions, and in this framework, the uncertainty of RUL predictions can be considered. Because of these developments, the model generalization for target-domain data is significantly improved. Regarding future research, adaptive threshold setting methods and the optimization method for model parameters and structures should be further developed.

Chapter 4: RUL prediction of machines operating under new conditions without corresponding training data

Unlike Chapter 3, this chapter considers the situation where no data can be collected from a machine operating under new conditions, rendering domain adaptation infeasible. To address this challenge, this chapter proposes a new prognostic model based on state-space modeling and reinforcement learning. The state-space model helps to quantify the discrepancies between actual observations and estimates for a specific operating parameter, thereby reducing the dependence of predicted degradation on working conditions. Furthermore, reinforcement learning allows for adversarial training to minimize prediction error under the most significant possible disturbance, enabling the designed model to be robust to potential operating conditions. The materials in this chapter are covered by the third research topic (Topic #3), which is introduced in Section 1.3. The organization of this chapter is as follows. In Section 4.1, an introduction to the reported methods on machine prognostics involving operating condition variations is made. The focused prognostic problem is formulated in Section 4.2. In Section 4.3, the proposed RUL prediction method based on state-space and neural network modeling is described. Two case studies are used to validate the proposed method in Section 4.4, including simulated gearbox signals and experimental bearing data. Lastly, conclusions are made in Section 4.5. The results of this chapter have been published as a journal paper [177].

4.1 Introduction

A machine generally operates under different conditions throughout its lifespan. For example, the bearing and gearbox in a wind turbine frequently experience fluctuating speeds and loads [121]. Despite recent advancements in prognostics, these variations in working conditions may have remarkable effects on the generalization of developed prognostic models, especially when the operating conditions are never encountered before.

In previous studies, various models have been proposed for RUL predictions, which have been discussed in detail in Section 1.2.2. These approaches can be divided into two main groups. In the

first group, the degradation of the machine is represented using state space models, which are based on stochastic processes such as the Gaussian model [42] or the Wiener process [43][44]. The degradation characteristics of machines, such as crack or wear size, are considered state variables, and their model parameters are determined using data-driven methods and historical condition data. Typically, one or two parameters follow a distribution to account for varying operating conditions, and these parameters are adjusted online based on different conditions using methods like Bayesian inference [39] and particle filter [40][41]. While state-space models are interpretable, they define RUL as the time until the machine degradation level reaches or exceeds a predetermined failure threshold. This approach assumes that the degradation mechanism remains the same across different operating conditions, and therefore, a subjective and fixed failure threshold can be set for all possible cases.

The second group of methods, as discussed in [51]-[68], focuses on the utilization of deep learning methods to construct prognostic models. The operating parameters are often considered as part of the input. By utilizing failure histories, a direct correlation between the condition monitoring signal and the corresponding RUL can be established. However, deep learning methods typically employ opaque black-box models, which may lead to subpar performance when faced with extrapolation problems caused by operating parameters that fall outside the range of recorded data. On the other hand, transfer learning can mitigate the disparities in monitoring signals by learning shared representations from various operating conditions. Ideally, a transfer learning-based prognostic model can leverage deep learning structures for domain adaptation and accurately predict the RUL for equipment operating under new conditions. However, domain adaptation methods necessitate access to condition monitoring data for the target operating conditions, even if this data is unlabeled.

The state-space model helps to measure the residuals between actual observations and estimates for a specific operating parameter. This reduces the reliance of the predicted degradation state on working conditions. However, when the state-space model of a machine degradation process is unknown and nonlinear, it becomes challenging to estimate its model parameters. In such cases, reinforcement learning is a useful tool as it can handle problems without a known state transition function. Reinforcement learning was first utilized for state-space identification in [122] and has been developed to show impressive performance in many later works [123]-[125]. However, for

unseen working conditions, large errors in the state-space model may cause the predicted RUL to converge in the wrong direction. To address this issue, this chapter proposes a new model-free prognostic method that combines state-space modeling and reinforcement learning. This method enables the prediction of RUL for machines operating under new conditions, even without corresponding historical signals.

The significant contribution of this study is twofold: 1) The Lyapunov constraint is incorporated into the reinforcement learning model. This means that during the training process, the estimated RUL for equipment will always asymptotically converge to the real RUL, providing a plausible explanation for the prognostic behavior of the model; 2) An adversarial training procedure is proposed based on the concept of $H\infty$ robustness. This helps to reduce the impact of errors in the state space on prognostic results. Overall, the proposed prognostic model is interpretable and capable of predicting the RUL for equipment operating in conditions that are not recorded.

4.2 Problem formulation

In data-driven prognostic methods, it is common to establish an offline nonlinear relationship between measurements and operating parameters and the corresponding RUL, which is used to predict the RUL online when new condition data is monitored. For example, if an LSTM model is used, the prognostic model can be constructed as:

$$r_t = LSTM(w_t, x_t, w_{t-1}, x_{t-1}, ..., w_{t-l}, x_{t-l}) \tag{4.1}$$

where l is the length of the preset time window, and x_t, w_t, and r_t represent the measurements, operating parameters, and the RUL at the inspection time t, respectively.

In general, if the learned RUL prediction model can work well online, the training dataset should record all possible operating conditions in the testing machine. However, in most cases, some new working conditions of equipment are not documented in the training dataset. According to the definitions in transfer learning [111], we define the training sample as “source-domain data” and the testing sample as “target-domain data”. It is assumed that parts of the operating conditions in the target domain are not recorded in the source domain. This study focuses on using only source-domain data to build a prognostic model for equipment in the target domain.

Let $D^S = \{D_1^S, D_2^S, ..., D_m^S\}$ denote *m* failure histories in the source domain, where $D_i^{(S)}$ is the *i*-th failure history with ls_i inspection points. $D_i^{(S)}$ is represented by $\{(x_{i,1}^S, w_{i,1}^S, r_{i,1}^S), (x_{i,2}^S, w_{i,2}^S, r_{i,2}^S), ..., (x_{i,lsi}^S, w_{i,lsi,}^S, r_{i,lsi}^S)\}$, and $(x_{i,t}^S, w_{i,t}^S, r_{i,t}^S)$ is the *t*-th inspection point of the *i*-th failure history, where $x_{i,t}^S$ represents monitoring measurements (or features), $w_{i,t}^S$ represents operating parameters, and $r_{i,t}^S$ is the corresponding RUL label. Similarly, let $D^T = \{D_{m+1}^T, D_{m+2}^T, ..., D_{m+n}^T\}$ represent *n* target-domain histories used to test the model performance. The *i*-th target-domain history has lt_i inspection points, denoted by $D_i^T = \{(x_{i,1}^T, w_{i,1}^T, r_{i,1}^T), (x_{i,2}^T, w_{i,2}^T, r_{i,2}^T), ..., (x_{i,lti}^T, w_{i,lti,}^T, r_{i,lti}^T)\}$, where $(x_{i,t}^T, w_{i,t}^T, r_{i,t}^T)$ is the *t*-th inspection point of the *i*-th target-domain history, and the dimension and measurement types of $x_{i,t}^T$, $w_{i,t}^T$ and $r_{i,t}^T$ are the same as the source-domain histories.

The problem of the present cross-domain prognostics can be formulated as: given enough failure histories in the source domain D^S, construct a nonlinear prognostic model to predict the RUL of equipment in the target domain D^T.

4.3 Proposed methodology

4.3.1 Overview

The key idea of the proposed method is to increase the sensitivity of the predicted RUL to unobserved degradation characteristics, e.g., cracks and wear, but decrease the sensitivity to operating parameters. Assume that the RUL will not significantly change between two inspection time points. This study will not directly construct a regression model like Eq. (4.1) between the condition data and their RUL. Instead, a nonlinear discrete-time state-space model is considered for the *i*-th failure history:

$$r_{i,t+1} = r_{i,t} + \varepsilon_{i,t} \tag{4.2}$$

$$x_{i,t+1} = f\left(g(r_{i,t+1}), w_{i,t}, x_{i,t}\right) + d_{i,t} \tag{4.3}$$

where, $x_{i,t+1}$ and $r_{i,t+1}$ are real measurements and RUL at time step t+1, $f(\cdot)$ and $g(\cdot)$ are nonlinear functions, $\varepsilon_{i,t}$ is the stochastic noise at time step *t*, and $d_{i,t}$ denotes the disturbance at time *t* caused by model errors and measurement noises. The RUL $r_{i,t}$ can be regarded as the state of the model and is modeled as a random walk [126]. The estimated measurements $\hat{x}_{i,t+1}$ at time step t+1

is modeled as $f(g(\hat{r}_{i,t}), w_{i,t}, x_{i,t})$, in which the term $g(r_{i,t})$ is modeled for the unobserved degradation characteristics at time *t*. The transition function Eq. (4.3) is a Markov process that only depends on the degradation characteristics $g(r_{i,t})$, the measurements, and operating parameters at present time *t*.

When the real measurements $x_{i,t+1}$ are observed at time step $t+1$ in the *i*-th failure history, the RUL can be predicted by using the following state estimator based on the idea of extended Kalman filter and unscented Kalman filter (UKF) [127]:

$$\hat{r}_{i,t+1} = \hat{r}_{i,t} + K_{i,t+1}\tilde{x}_{i,t+1} = \hat{r}_{i,t} + \Delta\hat{r}_{i,t+1} \tag{4.4}$$

where $\tilde{x}_{i,t+1} = x_{i,t+1} - \hat{x}_{i,t+1}$ is the estimated measurement error at time step $t+1$, and $\hat{r}_{i,t}$ and $\hat{r}_{i,t+1}$ denote the predicted RUL at time *t* and $t+1$, respectively. $K_{i,t+1}$ is the estimator gain at time $t+1$ that is calculated using partial derivatives of the function $f(\cdot)$ at $\hat{r}_{i,t}$, and thus the RUL increment $\Delta\hat{r}_{i,t+1}$ is a function of $\Delta\hat{r}_{i,t}$ and $\tilde{x}_{i,t+1}$.

Theoretically, if the disturbance in the model (4.3) is equal to zero, the measurement prediction error $\tilde{x}_{i,t+1}$ will not be sensitive to the operating parameters but sensitive to the degradation characteristics, since the estimated measurements $\hat{x}_{i,t+1}$ and the real measurements $x_{i,t+1}$ have the same operating parameters and different RULs. In addition, as shown in Fig. 4.1, the estimated measurements $\hat{x}_{i,t+1}$ of a machine are often governed by its physical model with different operating parameters *w*. Even though the training and testing data collected from a machine may experience different operating conditions, there is always only one function $Q(\hat{x}_{i,t+1} \mid w)$ between the operating parameters and the estimated measurements, which can be regarded as the dynamic physical model of the equipment. As a result, if the model (4.3) is well constructed and generalized, the predicted RUL will not be sensitive to operating parameters. And the proposed model also has the potential for RUL predictions of equipment in the target domain with different operating parameters.

Fig. 4.2 summarizes the procedure of the proposed method for RUL predictions. In this work, nonlinear functions $f(\cdot)$, $g(\cdot)$, and $\Delta\hat{r}_{i,t}$ are modeled as neural networks and trained using source-domain historical data D^S. In part (A), we use a multilayer perceptron (MLP) with the trick of dropout [128] to build functions $f(\cdot)$ and $g(\cdot)$. Although dropout can improve the generalization

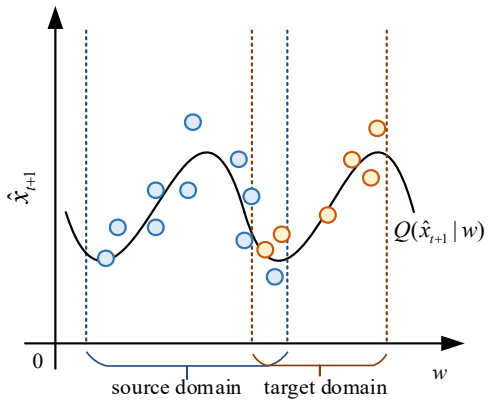


Fig. 4.1: Interpretation of the proposed method

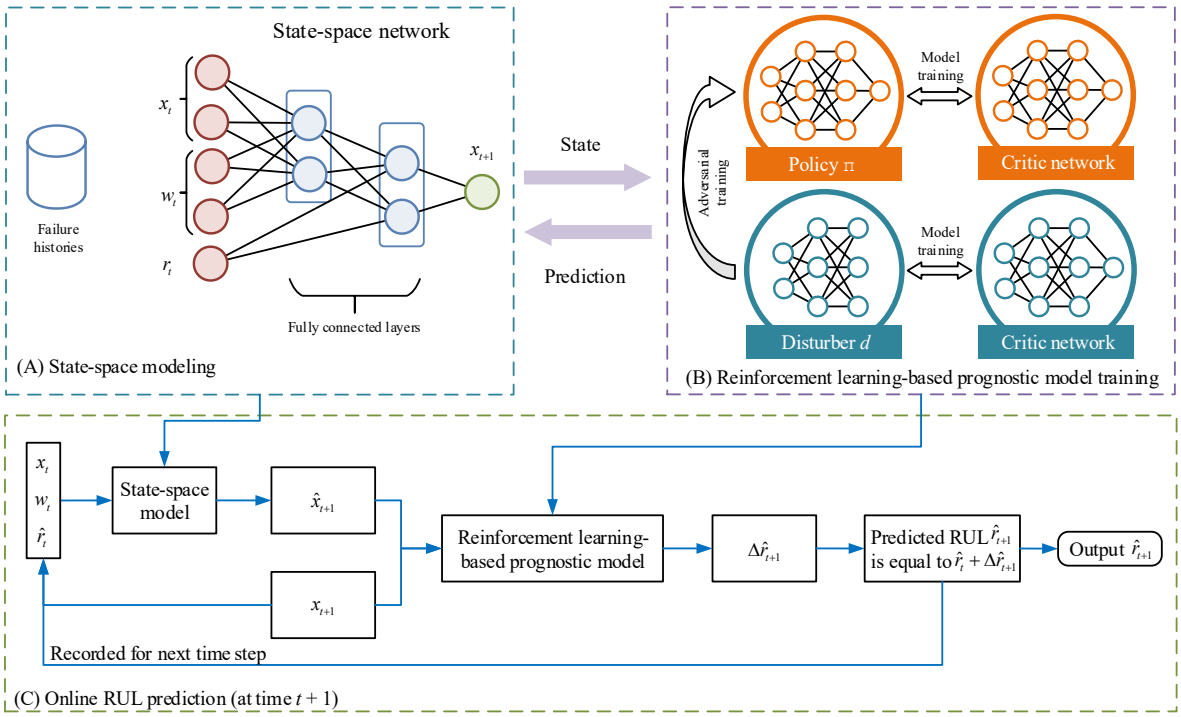


Fig. 4.2: Overview of the proposed prognostic method

of the MLP model in part (A), there will still be a large error between the learned state-space model and the real equipment physics, especially when the operating parameters are not documented in the training data. The most challenging part of using this prognostic framework is to predict $\Delta\hat{r}_{i,t}$ with unavoidable measurement estimation errors. Reinforcement learning is effective in identifying a function within a Markov decision process. This also allows for the consideration of the disturbance term in adversarial training to minimize prediction error under the largest possible

disturbance. Therefore, in part (B), $\Delta\hat{r}_{i,t}$ is trained as an MLP through reinforcement learning method based on the idea of Lyapunov stability and $H\infty$ robustness. Once each nonlinear function is learned, the model can be used for RUL prediction at each time point online in part (C) by calculating $\hat{x}_{i,t}$ and $\hat{r}_{i,t}$, iteratively. The model details in part (B) are introduced in Section 4.3.2, and the completed prognostic procedure is presented in Section 4.3.3.

4.3.2 RUL prediction based on reinforcement learning

In this study, $\Delta\hat{r}_{i,t}$ is approximated by a neural network $\pi(\cdot)$ with a deterministic output. First, it is assumed that the disturbance term $d_{i,t}$ in Eq. (4.3) is always zero. Given arbitrary initial model values, this study proposes a Lyapunov-based actor-critic model to ensure that the predicted RUL always asymptotically converges to the real RUL. Then, an adversarial training method based on the idea of $H\infty$ robustness is proposed to minimize the influence of the disturbance term on the RUL predictions.

Let $\tilde{r}_{i,t} = r_{i,t} - \hat{r}_{i,t}$ represent the RUL prediction error at time step t for the i-th failure history. The prognostic dynamics of the state estimator, i.e., Eq. (4.4), can be described as a Markov decision process (MPD) by Eq. (4.5), where $P(\cdot)$ is the state transition function.

$$\tilde{r}_{i,t+1} \sim P\left(\tilde{r}_{i,t+1} | \tilde{r}_{i,t}, \Delta\hat{r}_{i,t+1}\right) \tag{4.5}$$

Given a state $\tilde{r}_{i,t}$ as the initial state, the objective is to find an optimal $\pi^*(\cdot)$ to minimize the accumulated future RUL prediction error and future predicted RUL increments. In reinforcement learning algorithms, this objective can be defined as a Q-function [129] by Eq. (4.6). In this formulation, $C_{i,j}$ is the cost function at time step j in the i-th failure history, $\|\cdot\|_2$ denotes the second-order norm, γ_r is the discounting factor for $\pi(\cdot)$, and λ_r is the weighting factor.

$$Q^{\pi}(\tilde{r}_{i,t}, \Delta\hat{r}_{i,t}) = \sum_{j=t}^{ls_i} \gamma_r^j C_{i,j} \text{ , with } C_{i,j} = \left\|\tilde{r}_{i,j}\right\|_2^2 + \lambda_r \left\|\Delta\hat{r}_{i,j}\right\|_2^2 \tag{4.6}$$

Minimizing Eq. (4.6) is equivalent to minimizing the Bellman equation [129] by Eq. (4.7), where $\tilde{r}_{i,t+1}$ is the RUL prediction error at time step t+1 that can be calculated based on the state transition function Eq. (4.5) with an initial state $\tilde{r}_{i,t}$.

$$Q^{\pi}(\tilde{r}_{i,t}, \Delta\hat{r}_{i,t}) = C_{i,t} + \gamma_r Q^{\pi}(\tilde{r}_{i,t+1}, \Delta\hat{r}_{i,t+1}) \tag{4.7}$$

Motivated by the works in [124], in order to obtain a stable $\pi^*(\cdot)$, there will exist a Lyapunov function $L(\tilde{r}_{i,t})$ that satisfies $L(\tilde{r}_{i,t}) > 0$ and $L(\tilde{r}_{i,t+1}) - L(\tilde{r}_{i,t}) \leq 0$, so that the RUL prediction error will follow the direction of decreasing Lyapunov function values and eventually converge to zero. In the architecture of reinforcement learning methods, a Lyapunov candidate can be selected as the Bellman equation [124][130], i.e., Eq. (4.7). As a result, the constrained optimization problem can be constructed as:

$$\begin{aligned} &\min_{\pi} Q^{\pi}(\tilde{r}_{i,t}, \Delta\hat{r}_{i,t}) \\ &\text{s.t. } Q^{\pi}(\tilde{r}_{i,t+1}, \Delta\hat{r}_{i,t+1}) - Q^{\pi}(\tilde{r}_{i,t}, \Delta\hat{r}_{i,t}) \leq 0 \end{aligned} \tag{4.8}$$

Generally, reinforcement learning methods make use of two neural networks known as actor and critic, respectively. The actor network determines $\Delta\hat{r}_{i,t}$ based on the policy $\pi(\cdot)$, and the critic network evaluates the performance of actor network based on the value of $Q^{\pi}(\tilde{r}_{i,t}, \Delta\hat{r}_{i,t})$. In this work, two neural networks are trained using the source-domain data D^S. In order to satisfy the Lyapunov stability, the critic network is built as an MLP that relates the time-series input [$\tilde{r}_{i,t}$, $\Delta\hat{r}_{i,t}$] to the target output $Q^{\pi}(\tilde{r}_{i,t}, \Delta\hat{r}_{i,t})$. The actor network is built as an MLP with the input [$\Delta\hat{r}_{i,t-1}$, $\tilde{x}_{i,t-1}$, $\tilde{x}_{i,t}$] and the output $\Delta\hat{r}_{i,t}$, since $\Delta\hat{r}_{i,t}$ is a function of $\tilde{x}_{i,t}$ and $\Delta\hat{r}_{i,t-1}$ according to Eq. (4.4). The DNNs for actor and critic are illustrated in Fig. 4.3.

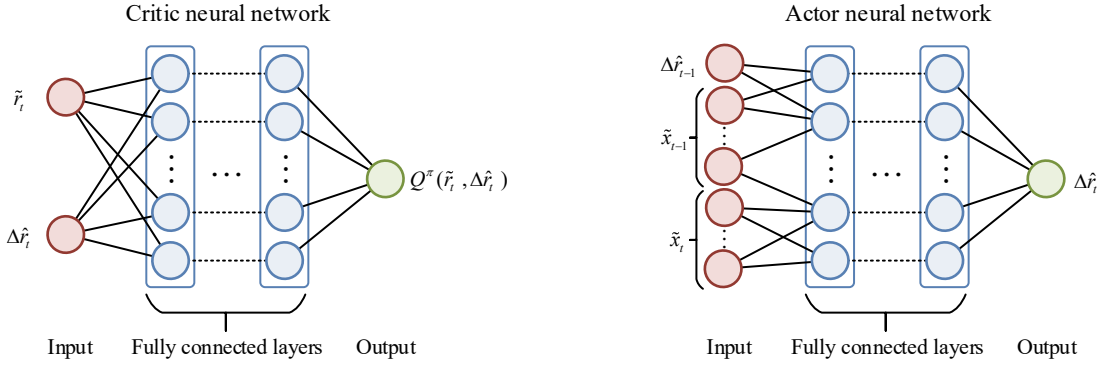


Fig. 4.3: Approximation of $\Delta\hat{r}$ using actor-critic networks (some connections are invisible)

During the model training, the transition of RUL prediction error is sampled according to Eq. (4.5), and the tuple ($\Delta\hat{r}_{i,t-1}, \Delta\hat{r}_{i,t}, \tilde{x}_{i,t-1}, \tilde{x}_{i,t}, \tilde{x}_{i,t+1}, \tilde{r}_{i,t}, \tilde{r}_{i,t+1}, C_{i,t}$) is stored in a dataset named replay memory

$\mathcal{M}_r$. The actor and critic are trained at each step by sampling a mini-batch uniformly from the replay memory. The loss functions of the critic and actor are defined as:

$$L_{critic} = \mathbb{E}_{\tilde{r}_{i,t},\tilde{r}_{i,t+1},\Delta\hat{r}_{i,t-1},\Delta\hat{r}_{i,t},\tilde{x}_{i,t-1},\tilde{x}_{i,t},\tilde{x}_{i,t+1},C_{i,t}\sim\mathcal{M}_r}\left[Q^{\pi}(\tilde{r}_{i,t},\Delta\hat{r}_{i,t}) - \left(C_{i,t} + \gamma Q^{\pi}(\tilde{r}_{i,t+1},\Delta\hat{r}_{i,t+1})\right)\right]^2 \tag{4.9}$$

$$L_{actor} = \mathbb{E}_{\tilde{r}_{i,t},\tilde{r}_{i,t+1},\Delta\hat{r}_{i,t-1},\Delta\hat{r}_{i,t},\tilde{x}_{i,t-1},\tilde{x}_{i,t},\tilde{x}_{i,t+1}\sim\mathcal{M}_r}\left[Q^{\pi}(\tilde{r}_{i,t},\Delta\hat{r}_{i,t}) + \max\left(Q^{\pi}(\tilde{r}_{i,t+1},\Delta\hat{r}_{i,t+1}) - Q^{\pi}(\tilde{r}_{i,t},\Delta\hat{r}_{i,t}),0\right)^2\right] \tag{4.10}$$

where, $\mathbb{E}(\cdot)$ is the expectation operator. The critic loss L_{critic} is defined by the temporal difference error [129], and the actor loss L_{actor} is trained to address the optimization problem in Eq. (4.8). The actor and critic networks are connected based on $\Delta\hat{r}_{i,t}$ and $\Delta\hat{r}_{i,t+1}$. Unlike the standard reinforcement learning methods, we add a penalty term in L_{actor} to make the learned model satisfy the Lyapunov constraint in Eq. (4.8). Under the Lyapunov constraint, the predicted RUL can always asymptotically converge to the real RUL with arbitrary initial values [125], thereby ensuring the training stability and interpreting the prognostic behavior of the proposed method.

In practice, there is always an error between the learned state-space model and the real physical system, especially when the equipment is faced with new operating conditions. This error is often caused by different operating parameters, degradation characteristics, and measurement noises. In this work, the error of the state-space model is considered as a disturbance term shown in Eq. (4.3). Since it is assumed that only the source-domain data is available for the model training, the learned disturbance cannot generalize well to the target-domain measurements. Therefore, our idea is to model a policy $\pi(\cdot)$ that is robust enough for the maximum possible disturbance, so that this policy will also be robust to disturbances in the target domain.

Based on the definition of $H\infty$ robustness [131], if a state-space model is stable when the disturbance $d_{i,t}$ in Eq. (4.3) is always zero, there will be a positive l_2 gain or $H\infty$ norm, so that the following holds for all possible $d_{i,t}$:

$$\int_0^{+\infty} C_{i,t}\,dt \le \int_0^{+\infty} \eta \left\| d_{i,t} \right\|_2^2 dt \tag{4.11}$$

where η is the $H\infty$ norm that quantifies the effect of disturbances on prognostic performance. A low value of the $H\infty$ norm implies that the designed functionality of the state-space model will be robust to possible disturbances.

In order to learn a policy $\pi(\cdot)$ with minimum $H\infty$ norm, we minimize the expected cost, i.e., Eq. (4.6), over the largest possible disturbance functions. The disturbance is modeled as a disturber $d(\cdot)$ by using actor-critic networks. The output of $d(\cdot)$ is designed as $d_{i,t}^{out} = [\mu_{i,t}]$, and its dimension is the same to the measurement dimension. At each time t, $d_{i,t}$ is sampled from a Normal distribution with mean $\mu_{i,t}$ and variance $\sum_{i,t} = \sigma_d^2 \mathrm{I}_d$, where I_d is the identity matrix with the dimension same to $\mu_{i,t}$, and σ_d is the preset standard deviation. Since the mean and variance of normal distribution will vary over time, the overall disturbances can follow arbitrary distributions. Let $s_{i,t}^d = [\hat{r}_{i,t-1}^d, \tilde{x}_{i,t-1}^d, \tilde{x}_{i,t}^d]$ denote the disturber state at time t in the i-th failure history, where $\hat{r}_{i,t-1}^d$ is the estimated RUL considering disturbance $d_{i,t-1}$ at time t-1, and $\tilde{x}_{i,t}^d = x_{i,t} - \hat{x}_{i,t} - d_{i,t}$ is the measurement prediction error considering disturbance at time step t. The actor-critic networks of disturber are illustrated in Fig. 4.4. The actor network is built as an MLP with the input $s_{i,t}^d$ and the output $d_{i,t}^{out}$. The critic network is built as an MLP that relates the input $[s_{i,t}^d, d_{i,t}^{out}]$ to a target output $Q^d(s_{i,t}^d, d_{i,t}^{out})$, and $Q^d(s_{i,t}^d, d_{i,t}^{out})$ is defined as the accumulated difference between RUL prediction error and possible disturbance:

$$Q^d(s_{i,t}^d, d_{i,t}^{out}) = \sum_{j=t}^{ls_i} \gamma_d^j C_{i,j}^d \text{, with } C_{i,j}^d = \left\| \tilde{r}_{i,j}^d \right\|_2^2 - \lambda_d \left\| d_{i,j} \right\|_2^2 \tag{4.12}$$

where $C_{i,j}^d$ is the disturber cost function at time step j in the i-th failure history, $\tilde{r}_{i,j}^d$ is the RUL prediction error considering disturbance at time step j, γ_d is the discounting factor for $d(\cdot)$, and λ_d is the weighting factor.

Similar to the actor-critic network, the tuple $(s_{i,t}^d, C_{i,t}^d, d_{i,t}^{out}, s_{i,t+1}^d)$ is stored in a replay memory $\mathcal{M}_d$ during the model training. The critic loss L_{critic}^d is defined by the temporal difference error, and the actor loss is defined to maximize the accumulated disturber cost $Q^d(s_{i,t}^d, d_{i,t}^{out})$. Therefore, the loss functions of the critic and actor in the disturber are defined as:

$$L_{critic}^d = \mathbb{E}_{s_{i,t}^d, s_{i,t+1}^d, d_{i,t}^{out}, C_{i,t}^d \sim \mathcal{M}_d} \left[Q^d(s_{i,t}^d, d_{i,t}^{out}) - \left(C_{i,t}^d + \gamma Q^d(s_{i,t+1}^d, d_{i,t+1}^{out}) \right) \right]^2 \tag{4.13}$$

$$L_{actor}^d = \mathbb{E}_{s_{i,t}^d \sim \mathcal{M}_d} \left[-Q^d(s_{i,t}^d, d_{i,t}^{out}) \right] \tag{4.14}$$

The disturber $d(\cdot)$ and the prognostic policy $\pi(\cdot)$ have opposing goals, so they can form an adversarial relationship to help the prognostic model find a policy $\pi(\cdot)$ with a minimum $H\infty$ norm, thus remaining robust to disturbances in the target domain.

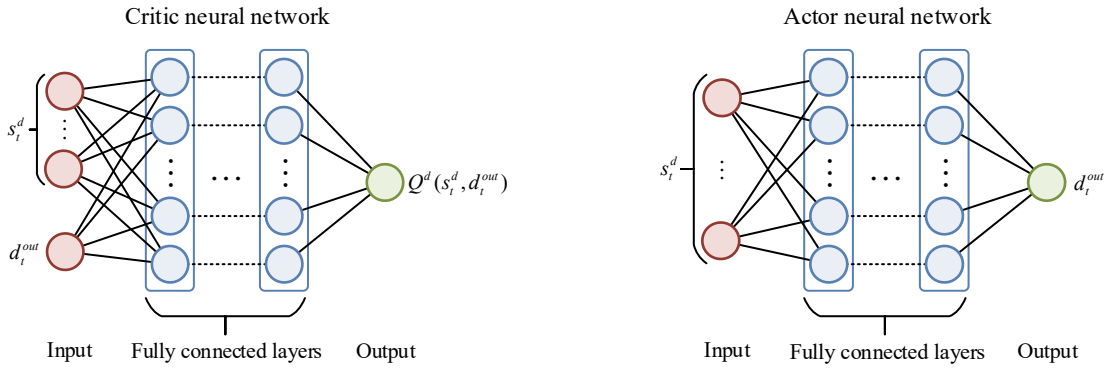


Fig. 4.4: Approximation of d^{out} using actor-critic networks (some connections are invisible)

4.3.3 Machine prognostics procedure

As shown in Fig. 4.2, the proposed prognostic models are trained offline and applied online iteratively. Given the failure histories $D_i^S = \{(x_{i,1}^S, w_{i,1}^S, r_{i,1}^S), (x_{i,2}^S, w_{i,2}^S, r_{i,2}^S), ..., (x_{i,lsi}^S, w_{i,lsi,}^S, r_{i,lsi}^S)\}$, i = 1, 2, …, m, t = 1, 2, …, ls_i, defined in Section 4.2, the offline modeling process is divided into two steps as follows.

Step 1: State-space modeling. MLP is selected to discover the mapping of $g(\cdot)$ and $f(\cdot)$ that relates the time-series input $[r_{i,t}^S, w_{i,t}^S, x_{i,t}^S]$ to a target output $x_{i,t+1}^S$, i = 1, 2, …, m, t = 1, 2, …, ls_i-1. The structure of the MLP is shown in Fig. 4.2. In the state-space model, $[w_{i,t}^S, x_{i,t}^S]$ is followed by fully connected layers and its end layer and $r_{i,t}^S$ are concatenated by fully connected layers to the output. The neuron size of each layer is similar to the measurement dimension. The dropout is added behind each hidden layer. 1/3 of the training data is used for the validation. Grid search is used to determine the hyperparameters, and the model with the minimum validation loss will be recorded.

Step 2: Prognostic model training. Algorithm 4.1 outlines the training procedure of the proposed prognostic model, where $C_{d,i,t}$ denotes the cost function defined in Eq. (4.6) considering disturbance at time step t in the i-th failure history. All basic models are constructed as MLPs, shown in Fig. 4.3 and Fig. 4.4. According to Occam's razor principle, all MLPs will use simpler

Algorithm 4.1: Proposed prognostic model based on reinforcement learning

Inputs: D^s, $\Delta\hat{r}_{max}$, μ_{max}, σ_d, γ_r, γ_d, λ_r, λ_d, N_π, N_d, batch size N_b, actor-critic network structure

Outputs: optimal prognostic policy $\pi^*(\cdot)$

For each iteration do
  for k = 1, 2,…, N_d
    for i = 1, 2,…, m, t = 1, 2, …, ls_i-1
      Store $(s_{i,t}^d, C_{i,t}^d, d_{i,t}^{out}, s_{i,t+1}^d)$ in $\mathcal{M}_d$; store $(\Delta\hat{r}_{i,t-1}^d, \Delta\hat{r}_{i,t}^d, \tilde{x}_{i,t-1}^d, \tilde{x}_{i,t}^d, \tilde{x}_{i,t+1}^d, \tilde{r}_{i,t}^d, \tilde{r}_{i,t+1}^d, C_{d,i,t})$ in $\mathcal{M}_r$; and store $(\Delta\hat{r}_{i,t-1}, \Delta\hat{r}_{i,t}, \tilde{x}_{i,t-1}, \tilde{x}_{i,t}, \tilde{x}_{i,t+1}, \tilde{r}_{i,t}, \tilde{r}_{i,t+1}, C_{i,t})$ in $\mathcal{M}_r$
      Sample N_b data from $\mathcal{M}_d$ to train $d(\cdot)$ according to Eq. (4.13) and Eq. (4.14)
    end for
  end for
  for k = 1, 2,…, N_π
    for i = 1, 2,…, m, t = 1, 2, …, ls_i-1
      Store $(s_{i,t}^d, C_{i,t}^d, d_{i,t}^{out}, s_{i,t+1}^d)$ in $\mathcal{M}_d$; store $(\Delta\hat{r}_{i,t-1}^d, \Delta\hat{r}_{i,t}^d, \tilde{x}_{i,t-1}^d, \tilde{x}_{i,t}^d, \tilde{x}_{i,t+1}^d, \tilde{r}_{i,t}^d, \tilde{r}_{i,t+1}^d, C_{d,i,t})$ in $\mathcal{M}_r$; and store $(\Delta\hat{r}_{i,t-1}, \Delta\hat{r}_{i,t}, \tilde{x}_{i,t-1}, \tilde{x}_{i,t}, \tilde{x}_{i,t+1}, \tilde{r}_{i,t}, \tilde{r}_{i,t+1}, C_{i,t})$ in $\mathcal{M}_r$
      Sample N_b data from $\mathcal{M}_r$ to train $\pi(\cdot)$ according to Eq. (4.9) and Eq. (4.10)
    end for
  end for
End for

structures and fewer neurons. All of the time-continuous failure histories are used to train the model. The output $\Delta\hat{r}_{i,t}$ is scaled from $-\Delta\hat{r}_{max}$ to $\Delta\hat{r}_{max}$, and the mean of disturbance $d_{i,t}$ is scaled from $-\mu_{max}$ to μ_{max}. These maximum scaled values can be defined based on the actual requirements and experiences. A high discounting factor γ means that the model loss will pay more attention to the cost at future time points. In this work, the factors γ_r and γ_d are set to constant values of 0.99 and 0.1, respectively. Weighting factors λ_r in Eq. (4.6), and λ_d in Eq. (4.12) can help the model focus its attention on specific tasks. In practice, we will increase the values of two weighting factors until the corresponding output $\Delta\hat{r}_{i,t}$ and $d_{i,t}$ will not always converge to the preset maximum value. During the model training, the prognostic policy $\pi(\cdot)$ and disturber $d(\cdot)$ are trained alternately by setting maximum iterations N_π and N_d. The parameters of one model are held constant during the training of another model. Our goal is to learn a policy $\pi(\cdot)$ that can work

stably with or without disturbances. Therefore, when the sum of training losses with and without disturbances is minimized, the model is recorded for the online RUL predictions.

In the online prognostic process, the measurements $\hat{x}_{t+1}$ are first estimated at time t based on the real measurements x_t, predicted RUL $\hat{r}_t$, and operating parameters w_t by the constructed state-space network $f(\hat{r}_t, w_t, x_t)$. When the real measurements x_{t+1} are observed at time t+1, the estimated measurement error $\tilde{x}_{t+1}$ is calculated. Afterward, the RUL increment $\Delta\hat{r}_{t+1}$ can be predicted by the learned optimal prognostic policy $\pi^*(\cdot)$ with input $[\Delta\hat{r}_t, \tilde{x}_t, \tilde{x}_{t+1}]$. Finally, the RUL at time t+1 can be predicted by $\hat{r}_{t+1} = \hat{r}_t + \Delta\hat{r}_{t+1}$, and the $\hat{r}_{t+1}$, $\Delta\hat{r}_{t+1}$ and $\tilde{x}_{t+1}$ are recorded for RUL predictions in the next time step.

4.4 Case studies

In this section, two examples are used to demonstrate and validate the benefits of the proposed method brought to RUL predictions. In Section 4.4.1, a simulation study with comparative analyses is conducted. In Section 4.4.2, an experimental bearing dataset is applied to evaluate the proposed model.

4.4.1 Case study 1: Simulated gearbox dataset

Based on the simulation model in [132] for generating gearbox vibration signals under time-varying speed conditions, we simulate a set of degradation signals randomly for comparative studies. For the i-th history, the vibration signal $y_{i,t}$ at time t is modeled as

$$y_{i,t} = \underbrace{\sum_{k_m=1}^{K_m} G_{k_m}(\omega_{i,t}) \cos\left(k_m 2\pi T_e \int_0^t \omega_{i,t} dt + \varphi_{k_m}\right)}_{\text{gear meshing vibration}} + \underbrace{\sum_{k_s=2}^{K_s} S_{k_s}(\omega_{i,t}) \cos\left(2\pi f_{k_s} t + \psi_{k_s}\right)}_{\text{structure vibration}} + \underbrace{D(\omega_{i,t}, \theta_{i,t}) S_d\left(t - T_d\right)}_{\text{defect impulses}} + \varepsilon_t \tag{4.15}$$

where t is the time step, $\omega_{i,t}$ is the rotating speed of input shaft at time t, $\theta_{i,t}$ is the degradation level at time t, T_e is the number of teeth, K_m is the number of gear meshing harmonics, K_s is the number of structure vibration components, φ_{km} is the initial phase for the k_m-th gear meshing harmonic, ψ_{ks} is the initial phase for the k_s-th structure vibration component, f_{ks} is the frequency of the k_s-th

structure vibration component, T_d is the time interval between two adjacent impulses, ε_t is the measurement noise at time t, $S_d(\cdot)$ is the defect impulse, and $G(\cdot)$, $S(\cdot)$, and $D(\cdot)$ are functions of $\omega_{i,t}$ to represent the amplitude of gear mesh, structure vibration, and impulses, respectively. To simulate the degradation characteristics of the gearbox, $D(\cdot)$ is modeled by a linear function related to a varying degradation term of the form $D(\omega_{i,t}, \theta_{i,t}) = 2\theta_{i,t} + \theta_{i,t}\omega_{i,t}$. The defect impulse is described by an exponentially decaying sinusoid as:

$$S_d(t) = e^{-\alpha t} \sin(2\pi f_r t) \tag{4.16}$$

where α is the decay rate, and f_r is the resonance frequency. The parameters are configured as listed in Table 4.1. The detailed parameter definition and selection of the model (4.15) can be referred to [132].

Table 4.1: Parameters configuration for the simulation model

Parameter	Value (s)	Parameter	Value (s)
T_e	37	ε_t	$\sim N(0, 1)$ m/s^2
Δt	0.001	K_s	2
K_m	4	ψ_1	$\pi/2$
φ_1	$\pi/2$	ψ_2	$\pi/2$
φ_2	$\pi/3$	$S_1(\omega_{i,t})$	$0.12\omega_{i,t}^2+0.5\omega_{i,t}+0.3$
φ_3	$\pi 2/3$	$S_2(\omega_{i,t})$	$0.2\omega_{i,t}^2+0.4\omega_{i,t}+0.2$
φ_4	$\pi 2/3$	f_1	140 Hz
$G_1(\omega_{i,t})$	$0.1\omega_{i,t}^2+0.3\omega_{i,t}+0.2$	f_2	180 Hz
$G_2(\omega_{i,t})$	$0.1\omega_{i,t}^2+0.3\omega_{i,t}+0.4$	f_r	140 Hz
$G_3(\omega_{i,t})$	$0.2\omega_{i,t}^2+0.6\omega_{i,t}+0.4$	α	100
$G_4(\omega_{i,t})$	$0.15\omega_{i,t}^2+0.5\omega_{i,t}+0.5$		

In addition, the degradation level is governed by the following model [133]:

$$\theta_{i,t_c+1} = \theta_{i,t_c} + abt_c^{b-1}e^{\alpha t_c^b} + \xi \tag{4.17}$$

where t_c is the degradation time cycle, a and b are degradation factors, and ξ denotes the linear degradation and process noise. Same as the [133], a is a value from 0.001 to 0.003, b is a value from 1.4 to 1.6, and ξ follows a Gaussian distribution with a mean of 0.001 and a variance of 0.002. Factors a and b are modeled as random Brownian motion.

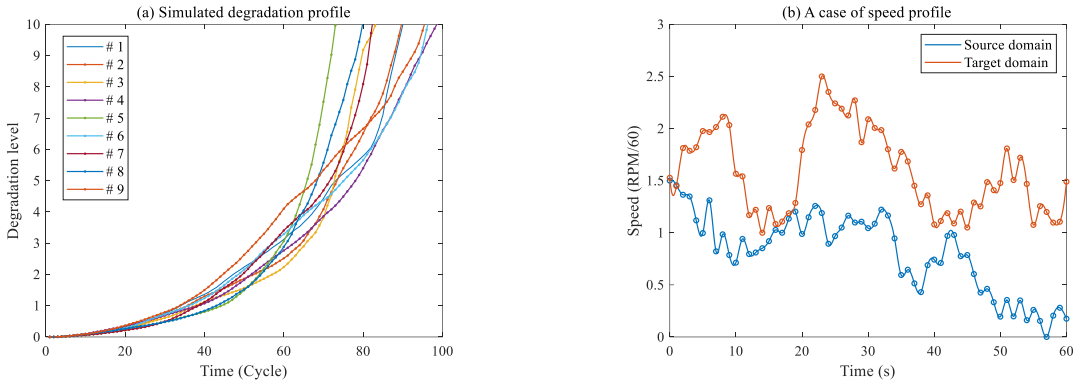


Fig. 4.5: Simulated degradation and speed profile

Table 4.2: Definition of extracted features

Time-domain	Frequency-domain	Time-frequency-domain
F1: RMS	F9: Mean of [0, 500]Hz	F14: Energy ratio of (3,0)
F2: Kurtosis	F10: Mean of [0, 125]Hz	F15: Energy ratio of (3,1)
F3: Peak-to-peak	F10: Mean of [126, 250]Hz	F16: Energy ratio of (3,2)
F4: Crest factor	F12: Mean of [251, 375]Hz	F17: Energy ratio of (3,3)
F5: Skewness	F13: Mean of [376, 500]Hz	F18: Energy ratio of (3,4)
F6: Shape factor		F19: Energy ratio of (3,5)
F7: Clearance factor		F20: Energy ratio of (3,6)
F8: Impulse factor		F21: Energy ratio of (3,7)

For one gearbox history, we assume that the gearbox completely fails if the corresponding degradation level reaches 10. In this study, nine gearbox failure histories are simulated, and their degradation trends are shown in Fig. 4.5 (a). For each failure history, 60 seconds vibration signals are simulated in one degradation cycle, and it is assumed that the degradation level will not change during these 60 seconds. Fig. 4.5 (b) shows two possible speed profiles in one cycle. The rotating speed of the first seven histories is set from 0 to 1.5, while the last two have the rotating speed from 1 to 2.5. The rotating speed is also simulated as random Brownian motion, and the continuous profiles can be obtained through cubic spline interpolation. The first five failure histories are used as the training data, and the last four histories are regarded as the testing data.

Every second, 1000 vibration measurements are generated. We extracted 21 features [15][52], shown in Table 4.2, from every 1000 measurements. Since a good prognostic feature should be monotonically correlated with the degradation process, we compute the correlation $Corr_i$ and the

monotonicity Mon_i according to [52] to evaluate the criteria coefficients for the i-th extracted feature:

$$Corr_i = \frac{\left|\sum_{j=1}^{n}(F_{i,j}-\bar{F}_i)(r_j-\bar{r})\right|}{\sqrt{\sum_{j=1}^{n}(F_{i,j}-\bar{F}_i)^2\sum_{j=1}^{n}(r_j-\bar{r})^2}} \tag{4.18}$$

$$Mon_i = \frac{\left|n_{dF_i>0}-n_{dF_i<0}\right|}{n-1} \tag{4.19}$$

where, n is the number of data points, and $F_{i,j}$ is the i-th feature value in the j-th data point, and r_j is the corresponding RUL. $\bar{F}_i$ and $\bar{r}$ are the mean of F_i and r. dF_i is the differential of the i-th feature series, and $n_{dF_i>0}$ means the number of data points with the differential of i-th feature larger than 0.

The criteria value Cri_i of the i-th feature is calculated as:

$$Cri_i = \frac{1}{2}\left(Corr_i + Mon_i\right) \tag{4.20}$$

The criteria coefficients of all features are normalized by min-max normalization. Ten features with a criteria value above a threshold of 0.5 are selected as the final input features, including F2, F3, F4, F5, F6, F7, F8, F9, F10, and F21. In addition, the integral of the rotating speed in one second is defined as the operating parameter of the corresponding features. Finally, we will have 26275 training data and 21356 testing data. All data are normalized to a value from 0 to 1 for comparative studies.

Four methods are used for comparative studies, including the common LSTM model, state-space modeling with UKF (SUKF) [126], state-space modeling with reinforcement learning model (SRL) proposed in Section 4.3.2, and the proposed state-space modeling with robust reinforcement learning method (SRRL). The last three methods use the same state-space model built by MLP, and both SRL and SRRL share the same actor-critic architecture and hyperparameters in the prognostic policy $\pi(\cdot)$. As compared to SRL, SRRL utilizes the proposed adversarial training to improve the model generalization. The detailed network architectures are briefly described below.

LSTM: The architecture of the LSTM comprises four layers. The network has two initial LSTM layers with 30 and 20 neurons plus dropout (p = 0.9) followed by a 10-way fully connected layer and ends with an output neuron. The network uses Tanh activation for each hidden layer and Sigmoid activation for the output layer. Grid search is used to determine the hyperparameters with the minimum validation loss. The time window size is set to 10.

State-space model in SUKF, SRL, and SRRL: The input [$w_{i,t}^S$, $x_{i,t}^S$] is followed by a fully connected layer with 5 neurons plus dropout (p = 0.9). This hidden layer is concatenated with $r_{i,t}^S$ and ends with a 10-dimensional output. The activation of hidden layers is Tanh function, and the output activation is Sigmoid activation. The hyperparameters are found by minimizing the validation loss.

Reinforcement learning model in SRL and SRRL: The actor of prognostic policy $\pi(\cdot)$ comprises two fully connected layers with 20 and 10 neurons and ends with a 1-dimensional output, and the activation of all layers are Tanh functions. The actor of disturber $d(\cdot)$ has a 30-way fully connected layer with Tanh activation and ends with a 10-dimensional output neuron with Tanh activation. Their critic networks comprise two fully connected layers with 30 and 20 neurons with Relu activations and end with a linear output neuron. The critic network of prognostic policy $\pi(\cdot)$ has two fully connected layers with 30 and 20 neurons with Relu activations and ends with a linear output neuron. Similarly, the critic network of disturber $d(\cdot)$ has a similar hidden structure, but 10 neurons are set in the second hidden layer. The hyperparameters $\Delta\hat{r}_{max}$, μ_{max} , σ_d , λ_r, λ_d, N_π, and N_d are set as 0.01, 0.2, 0.1, 100, 5, 8, and 5, respectively.

The root mean squared error (RMSE) and the mean absolute percentage error (MAPE) are used to evaluate the prognostic performance of each method:

$$RMSE = \sqrt{\frac{1}{n_t}\sum_{i=1}^{n_t}\left(RUL_{ture} - RUL_{predicted}\right)^2} \tag{4.21}$$

$$MAPE = \frac{1}{n_t}\sum_{i=1}^{n_t}\frac{\left|RUL_{ture} - RUL_{predicted}\right|}{RUL_{ture}},\ RUL_{ture} \neq 0 \tag{4.22}$$

where n_t is the number of the test data, and RUL_{true} and $RUL_{predicted}$ are the real and predicted RUL with the unit of a cycle, respectively. Since the testing data includes two domain histories, we use

$RMSE_S$ and $MAPE_S$ to represent the RMSE and MAPE when the source-domain testing data is used, and use $RMSE_T$ and $MAPE_T$ when the target-domain data is used.

In this case, each cycle has 60 samples with constant RUL. The operating conditions of the first two testing histories are similar to the training data, while the rest of the testing equipment works under conditions beyond the record. Fig. 4.6 shows the error between true and predicted RUL, i.e., $\Delta = RUL_{true} - RUL_{predicted}$, by comparative methods. In Fig. 4.6, the solid line represents the average RUL predictions for these 60 samples, and the shaded surface shows the variability of RUL predictions within each cycle. Overall, all methods get significantly better prognostic performance on the source-domain data than on the target-domain data, and the proposed SRRL outperforms the other three methods especially when the testing gearbox operates at speeds beyond the record. In addition, all predictions are very accurate for gearboxes close to failure. RUL estimates undertaken with LSTM at any time cycle have a large variability as compared to state-space-based approaches. SUKF shows less variability but a high bias compared to other methods, which means that the UKF algorithm cannot perform well when the state space is modeled by neural networks. SRL works well when the training and testing gearboxes operate at similar rotating speeds; however, overestimated RUL by SRL will lead to bad generalization for the target-domain data. In other words, adversarial training alleviates the sensitivity of the model to the disturbance caused by state-space errors so that SRRL can predict RUL for gearboxes operating under new conditions.

Fig. 4.7 and Fig. 4.8 show the RMSE and MAPE results obtained from 10 simulations using four methods with different random seeds, in which the range and standard deviation (SD) are used for descriptive error bars, and about two-thirds of the RMSE results will lie within the region of mean ± 1 SD, and ~95% of the RMSE results will be within 2 SD of the mean [108]. In Fig. 4.7 and Fig. 4.8, the middle line of each box denotes the mean value of 10 tests; the dark zone roughly denotes the 95% confidence interval of the mean value; the low transparency zone represents the region that contains about two-thirds of the data points. Between two boxes, the smaller the overlap of dark bars, or the larger the gap between dark bars, the smaller the statistical P-value and the stronger the evidence for a significant difference between the two methods [108]. It can be observed that SRRL performs better than the other three methods with respect to both source and target domain testing data, and shows statistically significant differences, especially between the $RMSE_T$ and $MAPE_T$ produced by other models. Compared to the LSTM-based method, SRRL

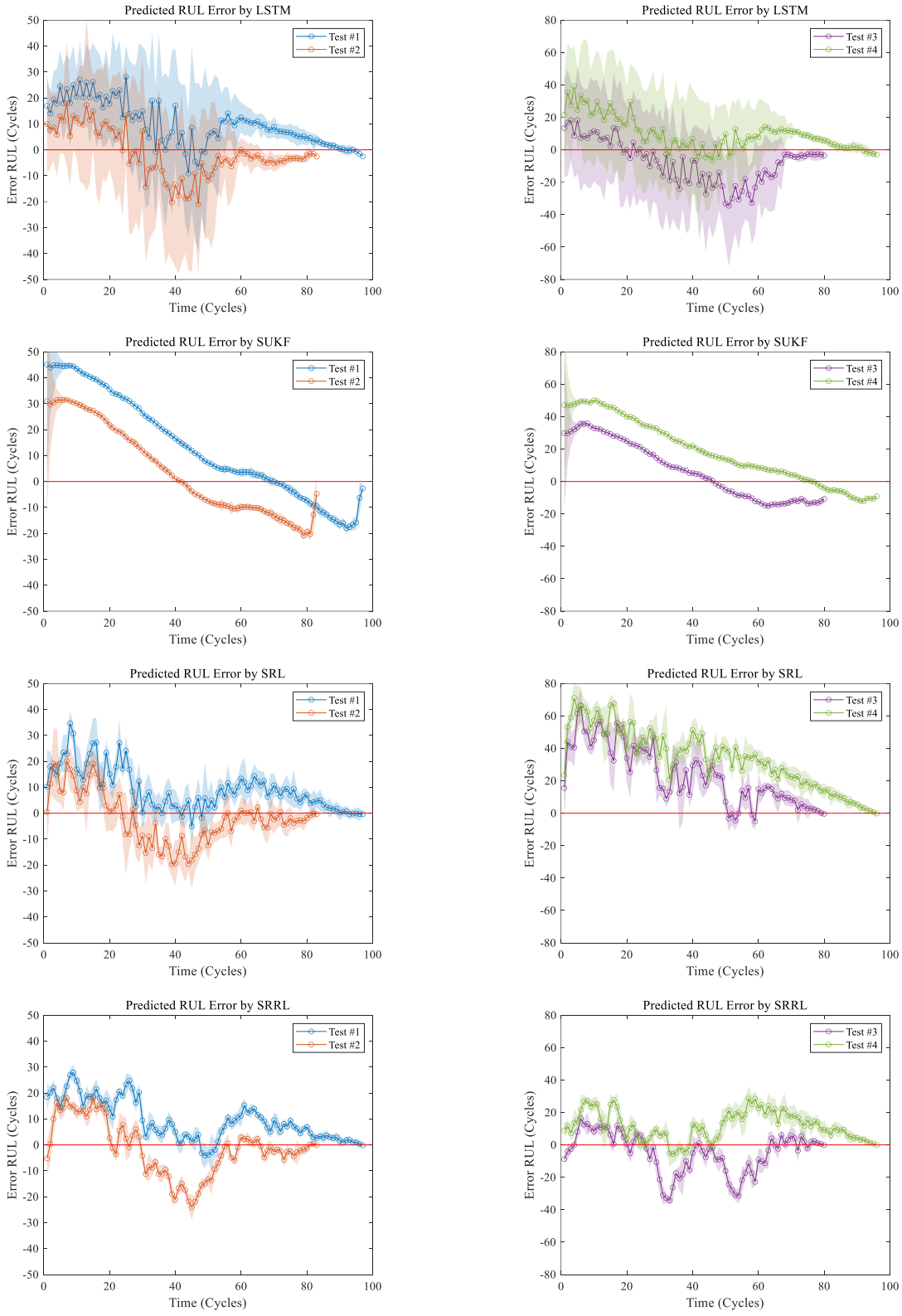


Fig. 4.6: RUL prediction error for each testing history by different methods

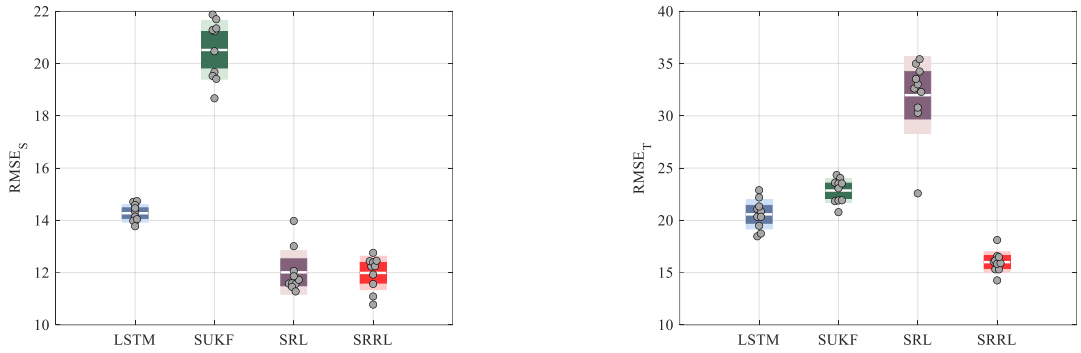


Fig. 4.7: RMSE of comparative methods using simulated gearbox dataset

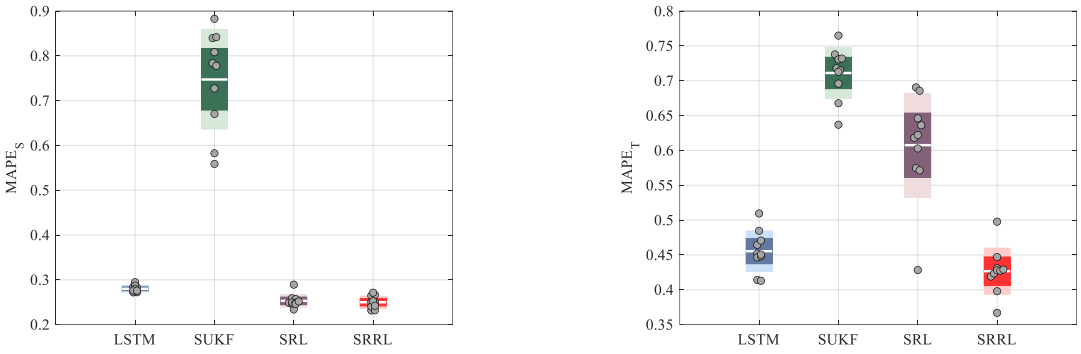


Fig. 4.8: MAPE of comparative methods using simulated gearbox dataset

reduces *RMSE* by about 20% and *MAPE* by nearly 10%, indicating that the proposed method has the potential to address the formulated cross-domain prognostic problem. In addition, although the differences in $RMSE_S$ and $MAPE_S$ between SRL and SRRL are not significant, SRL shows clear overfitting on the source-domain data and thus cannot generalize well to gearboxes operating in the target domain.

Moreover, the sensitivities from operating parameters and unobserved degradation characteristics to the predicted RUL are measured by linear multivariate Granger causality [134] with 10 lags. Table 4.3 presents the normalized causality measures for both LSTM and SRRL methods. The Granger causality method allows for a rough assessment of the causality between two factors. A high causality value implies that the predicted RUL by one method is significantly associated with a focused variable. Obviously, four tests undertaken with LSTM have large causalities from speed to predicted RUL, and small causalities from degradation level to RUL estimates as compared to SRRL. Especially for target-domain tests, SRRL shows a significant enhancement of causality

from degradation level to RUL estimates. The proposed SRRL increases the sensitivity of the predicted RUL to degradation characteristics and decreases the sensitivity to operating parameters by the state-space modeling. Therefore, SRRL is able to predict RUL for equipment working under target-domain conditions.

Table 4.3: Normalized causality from speed/degradation level to predicted RUL

	Test #1	Test #2	Test #3	Test #4
LSTM	0.0267/0.9733	0.1968/0.8032	0.1870/0.8130	0.1597/0.8403
SRRL	0.0600/0.9400	0.1385/0.8615	0.0681/0.9319	0.0420/0.9580

4.4.2 Case study 2: bearing dataset

In the second case study, the bearing dataset provided by the University of New South Wales (UNSW) is adopted [135]. The experiments are conducted on the test rig shown in Fig. 4.9. The dataset contains four different rotating speed conditions, including 6, 12, 15, and 20 Hz. At each speed condition, the bearing is repeated four times from health to failure. Therefore, a total of 16 run-to-failure bearing histories are collected. The sampling frequency of the vibration signal is 51.2 kHz. In this case, only the vertical vibration signal is used.

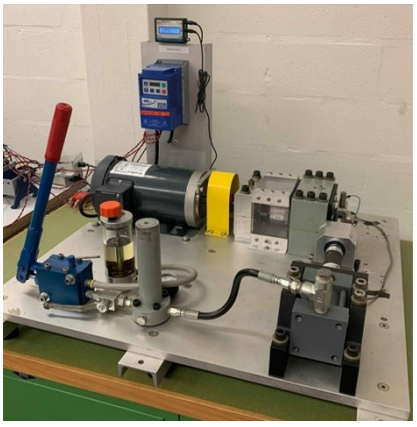


Fig. 4.9: UNSW experimental setup [135]

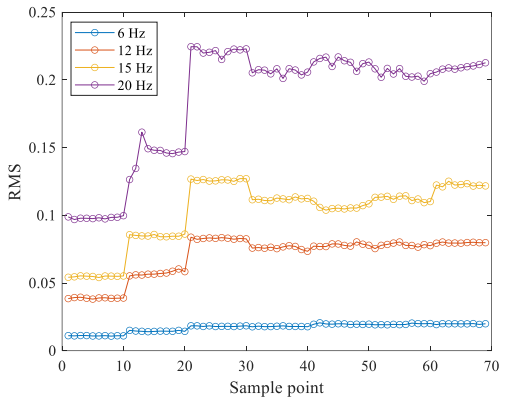


Fig. 4.10: RMS values of bearing with different speeds

A bearing is often inspected six to nine times during its lifetime. At each inspection, 12 s bearing vibration data are collected, and the middle 10 s vibration data are used to construct the dataset. The features in Table 4.2 are extracted from 51.2 k vibration data per second. Based on the evaluated criteria coefficients [52], six features with a normalized criteria value above 0.5 are selected as input features, including F1, F5, F6, F9, F10, and F13. We assume that the RUL will not change in one inspection. Finally, 10 sample points with the corresponding RUL percentage

labels from one to zero are obtained for each inspection cycle. In this case, the operating parameters contain both loads and speeds. Since the rotating speed is a constant value in one history data, a slight white noise is added before pre-processing. All data are normalized to a value from 0 to 1. As shown in Fig. 4.10, the same bearing with the same lifetime will present significantly different feature values at different rotating speeds. Twelve run-to-failure histories at 12, 15, and 20 Hz speeds (four histories per speed condition) are used as training data, and four histories at 6 Hz speed are used as testing data. Therefore, we have 870 data points for the model training and 290 data points for the testing.

Similar to the first case, four methods are used for the comparative study, and RMSE and MAPE are applied to evaluate their prognostic performance. The setting of components is described below.

LSTM: The architecture of the LSTM comprises three layers. The network has one initial LSTM layer with 30 neurons plus dropout (p = 0.9) followed by a 10-way fully connected layer and ends with an output neuron. The network uses Tanh activation for each hidden layer and Sigmoid activation for the output layer. The time window size is set to 5.

State-space model in SUKF, SRL, and SRRL: The input [$w_{i,t}^S$, $x_{i,t}^S$] is connected by a full layer with 5 neurons plus dropout (p = 0.9). This hidden layer is concatenated with $r_{i,t}^S$ and ends with a 6-dimensional output. The activation of hidden layers is Tanh function, and the output activation is Sigmoid activation.

Reinforcement learning model in SRL and SRRL: The actor of prognostic policy $\pi(\cdot)$ comprises two fully connected layers with 10 and 5 neurons and ends with a 1-dimensional output, and the activation of all layers are Tanh functions. Similarly, the critic network of disturber policy $\pi(\cdot)$ has a similar hidden structure but ends with a linear output neuron. The actor of disturber $d(\cdot)$ has a 30-way fully connected layer with Tanh activation and ends with a 6-dimensional output neuron with Tanh activation. The critic network of disturber $d(\cdot)$ comprises two fully connected layers with 30 and 10 neurons with Relu activations and also ends with a linear output neuron. The hyperparameters $\Delta\hat{r}_{max}$, μ_{max} , σ_d , λ_r, λ_d, N_π, and N_d are set as 0.3, 0.6, 0.15, 0.1, 1, 10, and 10, respectively.

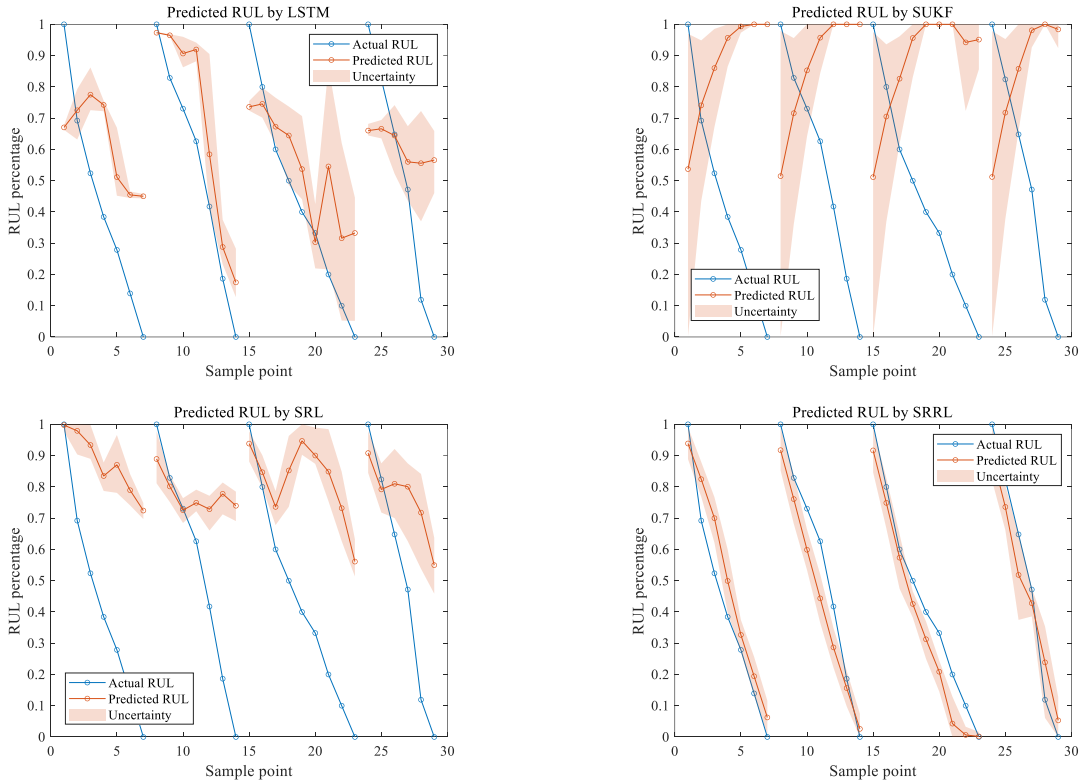


Fig. 4.11: RUL predictions for four test engines using different methods

Fig. 4.11 shows the RUL predictions for four testing bearings using four methods. In Fig. 4.11, the solid line, i.e., predicted RUL, represents the average RUL predictions for ten samples, i.e., 10 s data, and the upper and lower bounds of shaded surface, i.e., uncertainty, show the maximum and minimum predicted values for ten samples, respectively. It can be observed that RUL estimates undertaken with LSTM, SUKF, and SRL at any time cycle have high biases as compared to the proposed SRRL. Since the bearing features are significantly different at different rotating speeds, as presented in Fig. 4.10, the data-driven state-space model will have a large estimation error when dealing with bearings operating at new speeds. SUKF directly minimizes the residuals between the state-space model and observed signals, which may lead to convergence of the predicted RUL in the wrong direction. LSTM and SRL have similar RUL predictions with very high bias, indicating that the prognostic models fail to generalize to equipment with new operating conditions when the model only simply overfits training data. In contrast, the proposed SRRL method considers the state-space error as a disturbance term and uses adversarial training to reduce the influence of this error on RUL predictions, and thus the RUL estimated by SRRL are smooth and

monotone with less variability. SRRL shows a clear convergence trend, reflecting the importance of adversarial training in RUL predictions.

Fig. 4.12 presents the experimental results of RMSE and MAPE, in which each method is repeated 10 times with different random seeds and learning rates. In this case, the value of RMSE in Eq. (4.18) and MAPE in Eq. (4.19) are computed by the predicted and actual RUL percentage directly. Overall, the proposed SRRL outperforms the other three methods and shows a statistically significant difference in both RMSE and MAPE. With the use of SRRL, the reduction of RMSE ranges from 55% to 75% and MAPE ranges from 60% to 75% compared to other methods. Since only the proposed SRRL can successfully predict RUL for target-domain bearings, the improvement of SRRL in prognostics is quite significant. The sizes of low transparency zones in RMSE boxes of all comparative methods are similar, indicating that SRRL is also stable under consideration of different stochastic parameters.

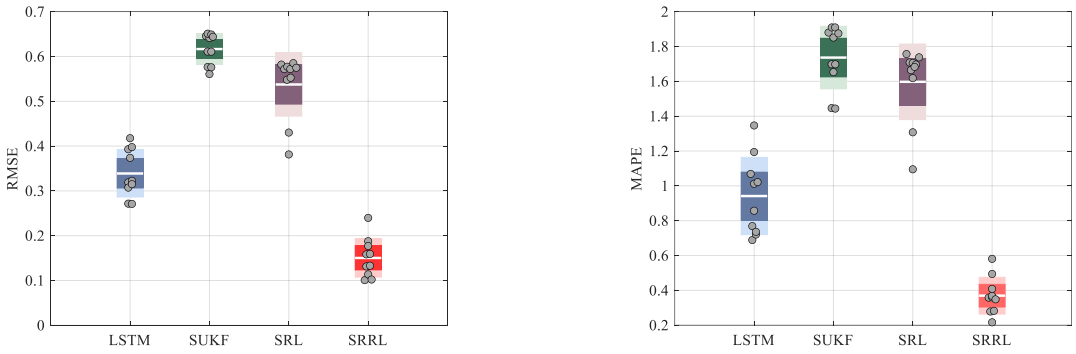


Fig. 4.12: RMSE and MAPE of comparative methods using experimental bearing dataset

4.5 Conclusions

This chapter proposes a novel prognostic method based on state-space and neural network modeling. Instead of predicting RUL by supervised-learning models, the proposed method identifies the system dynamics first and then estimates the RUL increment over time based on the state-space model. All of the nonlinear functions in this framework are modeled as neural networks and trained using failure histories. Based on the idea of Lyapunov stability and $H\infty$ robustness, a reinforcement learning model is proposed to interpret the prognostic behavior and predict RUL especially when the equipment operates under conditions beyond the record. The proposed

prognostic model is interpretable and can generalize well to other possible operating conditions. The conclusions from the case studies validate the accuracy of the proposed method.

The main contribution of this study is the construction of the time-series RUL prediction framework combined with state-space modeling and reinforcement learning methods. Meanwhile, the stability and robustness concepts are used to indirectly solve the error problem of the state-space model. This work is the first attempt to improve the generalization of the prognostic model for new extrapolated operating conditions in the absence of corresponding monitoring data (target-domain data). Compared to the approach discussed in Topic 2, which also considers RUL predictions across various conditions, this method is limited when the machine is significantly influenced by operating conditions. Moreover, if historical data includes suspended target-domain data, the model suggested in Topic 2 is easier to train and can deliver better performance. Regarding future research, suspension histories can be considered. In addition, sparse modeling methods can be attempted for state-space modeling to reduce the effect of identification errors.

Chapter 5: Condition-based maintenance optimization for multi-component systems considering prognostics and degraded working efficiency

Recent developments in prognostic models have led to increased use of prognostic information to improve the benefits of condition-based maintenance (CBM). However, existing CBM methods are often limited by the assumption that prognostic information is available for all components and by the neglect of economic loss due to system degradation in assessing benefits. As mentioned in Section 1.3, this chapter aims to find optimal maintenance decisions for multi-component systems with the consideration of partial prognostics and degraded working efficiency. The materials in this chapter are covered by the fourth research topic (Topic #4). The organization of this chapter is as follows. Section 5.1 introduces the reported studies on advanced maintenance policies. In Section 5.2, considered maintenance models are formulated, including the basic system model, prognostic errors, scheduled maintenance policies, and cost structures. In Section 5.3, the proposed system efficiency model and the definition of total net revenue are provided. The simulation procedure of maintenance actions and the optimization method are illustrated in Section 5.4. A case study of maintenance optimization on wind turbine farms is used to demonstrate the proposed method in Section 5.5. Finally, conclusions are made in Section 5.6. The results of this chapter have been published as a journal paper [178].

5.1 Introduction

With the progress in production processes and the increasing reliance on equipment, the expenses associated with system maintenance is growing. This has led to a growing demand for effective maintenance plans and actions [6]. As an illustration, more than a quarter of the overall costs throughout the lifecycle of offshore wind farms are allocated to maintenance operations [136]. In the chemical industry, maintenance activities engage around 30% of the workforce [7]. As industries move toward “Industry 4.0”, many companies are reevaluating their maintenance policies to stay competitive [9]. The advancements in sensor technology offer opportunities for the

development of maintenance strategies but also introduce new challenges for utilizing data in making maintenance decisions.

Maintenance policies used in asset management consist of three main types: corrective maintenance (CM), time-based maintenance (TBM), and condition-based maintenance (CBM). CM is a basic policy where maintenance actions are only taken when a component fails. On the other hand, TBM and CBM are preventive maintenance policies that determine maintenance actions based on constant inspection intervals and condition data, respectively. As discussed in Section 1.2.3, TBM is relatively easy to implement, as it only requires the age information of in-service components. However, one drawback of TBM is that it may lead to unnecessary maintenance actions when the machine is in good condition, resulting in extra maintenance costs. [14]. In contrast, CBM policies analyze condition data to make informed decisions about maintenance actions. This allows for more effective maintenance plans by targeting components that actually require attention [17].

Various methods for incorporating prognostics into maintenance optimization have been discussed in Section 1.2.3. When applying these advanced CBM policies to large mechanical systems, maintenance crews often take advantage of the opportunity to perform preventive maintenance on other components that have a relatively low predicted RUL whenever a component triggers the maintenance action. This is widely seen in studies [74][92][93]. The decision-making process for maintenance scheduling is done at the system level, with optimal maintenance actions planned based on key variables such as the planned inspection interval, preventive threshold, and opportunistic threshold. To find the most cost-effective decisions for maintenance, an optimization model can be formulated. This model aims to minimize the expected maintenance cost per unit of time, taking into account the maintenance decision variables. The optimization model is then integrated into a numerical simulation framework to consider practical factors in maintenance decision-making.

Based on the studies that have been reported, three limitations of the current prognostics-induced CBM policies are identified. First, these policies assume that condition monitoring can be carried out on all components in the system. Only one work considers a different situation but simply proposes independent maintenance strategies for different components in the system [137].

Additionally, while previous works mention that accurate RUL estimates can reduce routine maintenance by 40–70% [20], there is a lack of understanding on how the accuracy of the prognostic model influences the maintenance benefits, especially for systems with multiple components. Lastly, maintenance optimization centered on minimizing maintenance costs is gradually shifting towards maximizing profits over the planned operating period [138]. However, one limitation emerges when calculating profits in previous works, as they rarely take into account the impact of degradation on system profitability, despite its apparent significance in practice [4][5].

To address these gaps, this chapter proposes an approach that combines prognostics and CBM optimization. The approach has three main contributions: 1) It recognizes that not all components in the system can be continuously monitored. Therefore, the proposed maintenance optimization method aims to find a joint maintenance decision at the system level, including optimal planned inspection intervals and prognostic thresholds, under the assumption that only some components can be monitored. 2) The efficiency loss caused by system degradation is considered when evaluating the profitability of maintenance decisions. This is achieved by incorporating a data-driven efficiency modeling method into the net revenue objective function, with parameters estimated from historical system efficiency data. 3) Several practical factors are discussed in conjunction with the prognostics-induced CBM model. In particular, the model is extended with prognostic error modeling, allowing for insights into the impact of prognostic model accuracy on CBM benefits.

5.2 Formulation of the maintenance model

5.2.1 System description

Consider a multi-component system consisting of N sub-systems. Let $c_{i,j}$ represents the i-th component in the j-th sub-system, $i = 1, 2,\ldots, n_j$, and $j = 1, 2, \ldots, N$. Each sub-system contains n_j independent components. The failure of any component in a sub-system will lead to the sub-system failure. The failure of one sub-system will not cause the other subsystems to fail to work, but will result in a reduction in the benefit of the entire system. As an example, the studied system can be used to represent a wind farm with multiple wind turbines. Each wind turbine can be viewed as a sub-system consisting of multiple components, such as wind round, generator, and gearbox. Wind

turbines work independently of each other, but the failure of each turbine will result in economic loss to the wind farm.

We divide the components into two types. The degradation of both types of components can be assessed manually during planned inspection. The difference depends on whether the health status of the component can be continuously monitored. For the first type of component, the component can be monitored by sensors, so we can predict the RUL of these components in real time by prognostic models. For the second type of component, their health status can only be identified at each planned inspection or preventive maintenance since no sensors can be arranged to measure their conditions. In this work, the output of the prognostic model for component $c_{i,j}$ is defined as the RUL percentage denoted by $r_{i,j}$, which is a value from 0 to 1 equal to the quotient of its RUL divided by total lifetime. For some practical problems where the prognostic result is an RUL value, the proposed CBM optimization method will also work with a minor modification of the decision threshold variable. The health status is divided into multiple levels, and each level corresponds to an RUL percentage interval. Specifically, we define the last in-service health level corresponding to the RUL percentage from 0 to 0.05. We assume that the range of health status of components can be accurately identified during every inspection or preventive maintenance.

The goal of this work is to provide effective maintenance policies by maximizing production profits during the planned operating period. In real industries, the net revenue reflects the dynamics of two critical factors, i.e., production performance and maintenance cost [10]. To this aim, we define the expected net revenue as the difference between revenue and maintenance costs [138][139]. Let T and T_b denote the planned operating period of engineering assets and the baseline (most efficient) working time, respectively. The expected net revenue function during the planned operating period T is defined as:

$$R_{net} = \gamma T_b - C_{to} \tag{5.1}$$

Where γ is the reward rate at maximum efficiency, and C_{to} is the total maintenance cost. T_b represents the system operating time at the maximum efficiency, and thus T_b is equal to the product of T and system efficiency.

5.2.2 Modeling of prognostic errors

Various models are available from physics-based approaches to data-driven methods for RUL predictions. Compared to physics-based models, data-driven methods focus on using equipment histories to model a one-to-one relationship between condition monitoring data and the corresponding RUL so as to predict the RUL when new condition data is available. This work will not discuss the specific RUL prediction methods. Instead, a nonlinear function is used to represent the prognostic model by:

$$\hat{r}_{i,j}(t) = RUL(x_{i,j}(t), x_{i,j}(t-1), ..., x_{i,j}(t-l)) \tag{5.2}$$

where l is the length of the preset time window, and $x_{i,j}(t)$ and $\hat{r}_{i,j}(t)$ denote the condition measurements and predicted RUL percentage of the i-th component in the j-th sub-system at time t, respectively. $RUL(\cdot)$ is the nonlinear prognostic model, which can be constructed based on the equipment history data by many data-driven methods, such as long short-term memory (LSTM) [52] and convolutional neural network (CNN) [53].

Assume that the RUL of the component being monitored can be predicted at discrete interval Δt during the planned cycle T. Nonlinear prognostic models have RUL prediction errors during the online monitoring, and this error often converges to a value close to zero over time [126]. This implies that RUL predictions undertaken with prognostic models have large variability in early life as compared to the end of life. We use the mean-reverting Brownian process to model the difference between the real RUL and the predicted RUL, which has the property that the probability distribution of prognostic errors is stationary over time. However, the mean-reverting Brownian is an asymptotic convergence process and thus cannot reflect the random positive and negative fluctuations of the error. Therefore, prognostic errors are generated by the following procedures.

First, for the component $c_{i,j}$ with RUL percentage $r_{i,j}$ from 0 to 1, its prognostic errors from RUL percentage $r_{i,j} = 1$ to $r_{i,j} = 0$ are pre-generated by mean-reverting Brownian process:

$$Err_{i,j}(1 - r_{i,j} + \Delta r_{i,j}) = Err_{i,j}(1 - r_{i,j}) + \theta_e \left(\mu_e - Err_{i,j}(1 - r_{i,j}) \right) \Delta r_{i,j} + \sigma_e Y \sqrt{\Delta r_{i,j}} \tag{5.3}$$

where $Err_{i,j}(1- r_{i,j})$ is the prognostic error of the i-th component in the j-th sub-system when its RUL percentage is $r_{i,j}$, θ_e is the rate of convergence to the mean, μ_e is the mean value of the process, σ_e denotes the diffusion parameter characterizing the error variation, and Y follows a standard Normal distribution with 0 mean and variance of 1. We define the value of μ_e to be zero to indicate that the predicted RUL can always converge to the real RUL.

The prognostic error $Err_{i,j}(1-r_{i,j})$ can be either a positive or a negative number. Therefore, $Err_{i,j}(1-r_{i,j})$ will be randomly equal to its positive or negative value, and we can get the predicted RUL percentage $\hat{r}_{i,j}$ equal to $r_{i,j} - Err_{i,j}(1-r_{i,j})$. Then, the predicted RUL percentage $\hat{r}_{i,j}$ is bounded between 0 and 1. If the calculated $\hat{r}_{i,j}$ is larger than 1 or smaller than 0, its value will be scaled to 1 or 0 correspondingly. Finally, $Err_{i,j}(1-r_{i,j})$ can be obtained by $r_{i,j} - \hat{r}_{i,j}$.

In Eq. (5.3), the value of $Err_{i,j}(1-r_{i,j})$ depends on the initial value $Err_{i,j}(1)$ and model parameters θ_e and σ_e. To measure the effect of prognostic errors, a widely used indicator, i.e., the root mean squared error (RMSE), is estimated for the component $c_{i,j}$ as:

$$RMSE_{i,j} = \sqrt{\Delta r_{i,j} \sum_{r_{i,j}=0}^{1} Err_{i,j}(1-r_{i,j})^2} \tag{5.4}$$

Commonly, the RMSE of prognostic models varies from 0.05 to 0.2 [44]. Obviously, if RMSE = 0, then the prognostic model is perfect. Since the prognostic error cannot be generated according to a specific RMSE value, we first generate several prognostic errors with stochastic initial value and model parameters θ_e and σ_e following the procedure described above and save the errors that satisfy the RMSE range requirement in a database D_e. In this way, we can randomly generate as many prediction errors as possible for all possible ranges of RMSE. Thereby, the prognostic error can be sampled directly for a specific RMSE range from the database.

5.2.3 Maintenance scheduling and cost modeling

In this work, corrective replacement, planned inspection, preventive replacement, and opportunistic replacement are designed in our prognostics-induced CBM policy. We perform preventive replacement based on the components' predicted RUL. Since not all components can be monitored, planned inspections are scheduled to prevent the failure of non-monitorable components. For the considered system that has multiple subsystems and components, performing

maintenance simultaneously on other components with relatively low RUL will yield lower maintenance costs than maintaining them individually, and thus, opportunistic replacement is also considered. There are three decision variables that control maintenance frequencies, including planned inspection interval t_m, preventive threshold p_{pm}, and opportunistic threshold p_{om}. The length of time between planned inspections is usually variable. Longer inspection intervals are scheduled early in the period, while shorter intervals are determined later. Let n_m denote the number of planned inspections, i.e., $\sum_{i=1}^{n_m} t_{m,i} = T$. Oftentimes, asset management expects to use a uniform maintenance policy for a system, and therefore, we assume that the inspection intervals and maintenance thresholds are the same for different components. The maintenance policies are outlined as follows:

- Corrective replacement: If the component $c_{i,j}$ fails, CM is executed.
- Planned inspection: If the inspection time is reached, planned inspection is performed, and the component $c_{i,j}$ is replaced if it reaches the last in-service health level, i.e., RUL percentage from 0 to 0.05.
- Preventive replacement: If the component $c_{i,j}$ can be monitored in real time, preventive replacement is performed for component $c_{i,j}$ when the predicted RUL percentage $\hat{r}_{i,j}$ of component $c_{i,j}$ is less than p_{pm}.
- Opportunistic replacement: Opportunistic replacement of component $c_{i,j}$ in the j-th sub-system is accepted when any of the following conditions is met: (a) any other component in the j-th sub-system is determined to be maintained, and the component $c_{i,j}$ reaches the last in-service health level with RUL percentage from 0 to 0.05, or (b) any other sub-system k, $k = 1, \ldots, j-1, j+1, \ldots, N$, fails or is preventively maintained, the component $c_{i,j}$ can be monitored in real time, and the predicted RUL percentage $\hat{r}_{i,j}$ of component $c_{i,j}$ is less than p_{om}.

Generally, opportunistic replacement should be executed before the preventive replacement for meaningful solutions. Therefore, two RUL percentage thresholds satisfy $p_{pm} < p_{om}$, which will be added as a constraint in the model solving. Opportunistic replacement is effective when the operation of each sub-system has an impact on the benefits of the whole system. Fig. 5.1 is

presented to illustrate the provided opportunistic replacement. When corrective or preventive replacement is performed on a sub-system, the maintenance crew will take this opportunity to perform maintenance on other qualified components. Taking three components, i.e., components 1, 2, and 3, as an example, components 1 and 2 are in one sub-system being maintained, while component 3 is in the other working sub-system. When a sub-system is maintained, the health status of all its components can be identified, and if a component reaches the last in-service health level, i.e., the real RUL percentage is less than 0.05, that component will be replaced. For other working sub-systems, the component will be replaced if the predicted RUL percentage of one monitored component is less than the opportunistic threshold p_{om}. Therefore, opportunistic replacement is accepted for components 1 and 2 but rejected by component 3.

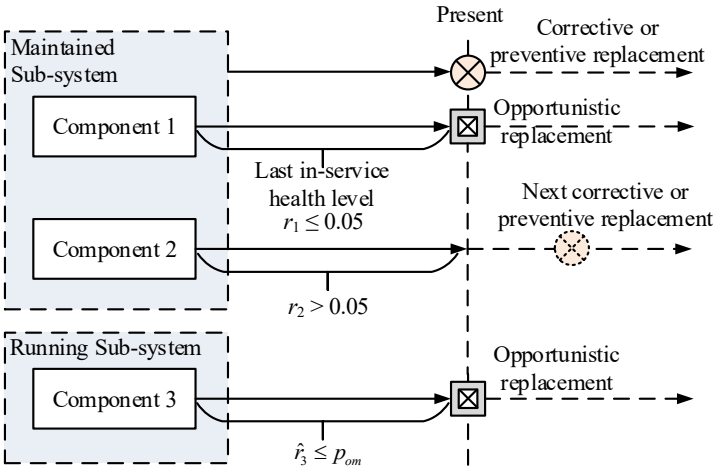


Fig. 5.1: Illustration of the opportunistic replacement

In practice, each policy will have a component-dependent replacement cost and a setup cost. Three strategies use the same replacement cost for each component. The setup cost is common and shared for three strategies due to scaffolding, crew traveling, shutdown, etc. The setup cost of opportunistic and preventive replacements is often equal but much lower than corrective replacement. The following cost structure is considered in the maintenance cost model:

- $C_{r,i}$: Replacement cost for the i-th component when component replacement is required.
- C_{fs}: Setup cost when corrective replacement is prepared.
- C_{ps}: Setup cost when opportunistic or preventive replacement is prepared.

In single maintenance, if multiple strategies are executed simultaneously, only one and the highest setup cost is considered. We denote the cost of the *k*-th maintenance cost during the planned period *T* by $C_{m,k}$. With the scheduled cost structure, suppose that preventive, opportunistic, and corrective replacements are performed simultaneously and three components, i.e., components 1, 2, and 3, are determined to be replaced in the *k*-th maintenance, the setup cost will only incur once, and the maintenance cost $C_{m,k}$ will be $C_{r,1} + C_{r,2} + C_{r,3} + C_{fs}$. With the same maintenance plan, if corrective replacement is not executed, the maintenance cost $C_{m,k}$ will be $C_{r,1} + C_{r,2} + C_{r,3} + C_{ps}$.

Let N_m represent the total number of maintenance executed during the planned period *T*, the total maintenance cost can be formulated as follows:

$$C_{to} = \sum_{k=1}^{N_m} C_{m,k} \tag{5.5}$$

5.3 Net revenue estimation considering degraded working efficiency

In this section, the degradation-related system efficiency is incorporated into the net revenue model to measure maintenance performance. The proposed approach starts with the modeling of working efficiency loss for each component. Efficiency losses are modeled by the Wiener process, and model parameters are determined at the system level based on historical data. A new net revenue function is then proposed to consider the degraded working efficiency in maintenance decision-making. Accordingly, the maintenance decisions can be optimized in Section 5.4.

5.3.1 Degraded working efficiency modeling at component level

This work uses the loss of efficiency to measure the degraded working efficiency. The efficiency loss at time *t* is defined as the ratio of the reduced efficiency up to time *t* to the initial efficiency. As reported in [140], the Wiener process has outstanding potential in modeling the efficiency trends of machines by allowing for the inevitable efficiency fluctuations in the degradation process. Therefore, in this work, the system efficiency loss due to the degradation of component $c_{i,j}$, denoted by $L_{i,j}$, is modeled by a Wiener process:

$$L_{i,j}(1-r_{i,j}) = L_{i,j}(0) + \theta_{i,j}(1-r_{i,j})^{a_{i,j}} + \sigma_{i,j}B(1-r_{i,j}) \tag{5.6}$$

where $L_{i,j}(1–r_{i,j})$ denotes the system efficiency loss caused by the component $c_{i,j}$ with RUL percentage $(1–r_{i,j})$, $B(1–r_{i,j})$ represents the standard Brownian motion, and $\theta_{i,j}$ and $\sigma_{i,j}$ are the drift parameter and the diffusion parameter, respectively. The value of $r_{i,j}$ decreases from 1 to 0 during the lifetime of the component. $L_{i,j}(0)$ equals 0 at the beginning of the components' lifetime. When $a_{i,j} = 1$, the Wiener degradation model is linear.

The relationship between the system efficiency losses caused by the component $c_{i,j}$ at RUL percentage $r_{i,j}$ and at RUL percentage $(r_{i,j}–\Delta r_{i,j})$ can be expressed as Eq. (5.7), where Y follows the standard Normal distribution.

$$L_{i,j}(1-r_{i,j}+\Delta r_{i,j}) = L_{i,j}(1-r_{i,j}) + \theta_{i,j} a_{i,j} \Delta r_{i,j} (1-r_{i,j})^{a_{i,j}-1} + \sigma_{i,j} Y \sqrt{\Delta r_{i,j}} \tag{5.7}$$

According to the statistical property of the Wiener process, $L_{i,j}(1–r_{i,j})$ also follows a Normal distribution as:

$$L_{i,j}(1-r_{i,j}) \sim N(\theta_{i,j}(1-r_{i,j})^{a_{i,j}}, \sigma_{i,j}^2(1-r_{i,j})) \tag{5.8}$$

It is assumed that the system efficiency loss caused by the degradation of component $c_{i,j}$ reaches approximately $d_{i,j}$ when the corresponding RUL percentage reduces to 0, that is, the average value of $L_{i,j}(1)$ is approximately equal to $d_{i,j}$. Based on Eq. (5.8), we can get $\theta_{i,j} = d_{i,j}$ for the component $c_{i,j}$.

5.3.2 Parameter estimation of working efficiency model at system level

Let $D = \{D_1, D_2, ..., D_M\}$ denote M history data, where D_m, $m = 1,\ldots, M$, is the m-th history with $l_{f,m}$ time points, represented by $D_m = \{(t_{m,1}, e_{m,1}), (t_{m,2}, e_{m,2}), ..., (t_{m,lfm}, e_{m,lfm})\}$, and $(t_{m,n}, e_{m,n})$ is the n-th recording point of the m-the history, where $e_{m,n}$ is the total system working efficiency and $t_{m,n}$ is the corresponding recording time. It is worth noting that the working efficiency of the system can be represented by different records, such as average production or power. History data is further processed, then we can get $D_m = \{(t_{m,1}, L_{m,1}), (t_{m,2}, L_{m,2}), ..., (t_{m,lfm}, L_{m,lfm})\}$, where $L_{m,n} = (e_{m,1} - e_{m,n}) / e_{m,1}$ is the system efficiency loss at the time $t_{m,n}$ in the m-the history. Commonly, $L_{m,1}$ is equal to 0, indicating that the system has the maximum expected working efficiency at the beginning of the lifetime.

At the system level, the influence of the structural dependency of components on the system efficiency loss can commonly be modeled by the normal copula function with a defined correlation matrix [141]. The system efficiency loss can be estimated as a linear sum of the efficiency losses due to the degradation of all components, which are sampled from the copula function. Simplistically, we neglect the effect of structural dependencies on the system efficiency loss in this work to demonstrate the parameter estimation approach. According to Eq. (5.8), the system efficiency loss at time t follows a Normal distribution as Eq. (5.9), where $r_{i,j,t}$ denotes the RUL percentage of the component $c_{i,j}$ at time t.

$$L(t) = \sum_{j=1}^{N}\sum_{i=1}^{n_j} L_{i,j}(r_{i,j,t}) \sim N(\sum_{j=1}^{N}\sum_{i=1}^{n_j} \theta_{i,j}(1-r_{i,j,t})^{a_{i,j}}, \sum_{j=1}^{N}\sum_{i=1}^{n_j} \sigma_{i,j}^2(1-r_{i,j,t})) \tag{5.9}$$

Let $\mu_{L,t} = \sum_{j=1}^{N}\sum_{i=1}^{n_j} \theta_{i,j}(1-r_{i,j,t})^{a_{i,j}}$ and $\sigma_{L,t} = \sqrt{\sum_{j=1}^{N}\sum_{i=1}^{n_j} \sigma_{i,j}^2(1-r_{i,j,t})}$. The probability density function (PDF) of $L(t)$ is given by:

$$f(L(t)) = \frac{1}{\sigma_{L,t}\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{L(t)-\mu_{L,t}}{\sigma_{L,t}}\right)^2} \tag{5.10}$$

To find out the difference in system efficiency loss due to the degradation of different components, the analytic hierarchy process [142] is used to evaluate the weight of the impact of each component degradation on the system efficiency. Then, we can get the weight of each component $w_{i,j}$, with $\sum_{j=1}^{N}\sum_{i=1}^{n_j} w_{i,j} = 1$. For any two components in the j-th sub-system (taking components 1 and 2 as an example), their drift parameters satisfy $\theta_{1,j} / \theta_{2,j} = w_{1,j} / w_{2,j}$, which implies that we can define the drift parameter of only one component and determine parameters for other components based on their weights.

Let parameter set $\Omega = \{\theta_{1,j}\,(j = 1,\ldots, N), a_{i,j}\,(i = 1,\ldots, n_j, j = 1,\ldots, N), \sigma_{i,j}\,(i = 1,\ldots, n_j, j = 1,\ldots, N)\}$. According to Eq. (5.10), the log-likelihood function of $L(t)$ can be expressed by Eq. (5.11). In this formulation, t_{max} is the maximum system life in the history data, and $n_{L,t}$ is the number of system efficiency loss data L at time t.

$$\ln L(\Omega \mid L(t)) = \frac{1}{t_{\max}}\sum_{t=0}^{t_{\max}} \frac{1}{n_{L,t}} \sum_{k=1}^{n_{L,t}} \ln f(L(t)) \tag{5.11}$$

Oftentimes, the lifetime distributions of all components are known in advance, which can be estimated using failure time and censored time data. When a set of failure times is sampled from the lifetime distribution of all components, the RUL percentage r_t of all components at a specific time t can be calculated, and the μ_L and σ_L can be obtained. Therefore, based on the idea of Monte Carlo simulation (MCS), we sample n_s sets of components' failure times for one system efficiency loss data, and the log-likelihood function of $L(t)$ is formulated as:

$$\ln L(\Omega \mid L(t)) = \frac{1}{t_{\max}} \sum_{t=0}^{t_{\max}} \frac{1}{n_{L,t}} \sum_{k=1}^{n_{L,t}} \frac{1}{n_s} \sum_{l=1}^{n_s} \ln f(L(t)) \tag{5.12}$$

Finally, the optimal maximum likelihood estimator Ω^* can be obtained by maximizing the log-likelihood function $\ln L(\Omega \mid L(t))$ with the use of global optimization algorithms, such as the genetic algorithm (GA). Since the model parameters are determined from the historical data, it is usually possible to consider the impact of some realistic factors, e.g., environmental and maintenance actions, on the loss of system efficiency. In addition, the proposed parameter estimation procedure is also feasible when structural dependencies are taken into account and defined by a correlation matrix, since the μ_L and σ_L can be calculated based on MCS by sampling from the copula function. In the parameter estimation, the parameter $a_{i,j}$, $i = 1,\ldots, n_j$, $j = 1,\ldots, N$, will be scaled from 1 to 2 for reasonable solutions.

5.3.3 Net revenue modeling

During the planned operating period T, the operating time can be discretized into multiple time intervals based on a preset Δt. Considering the system working efficiency loss $L(t)$, the effective working time at time t can be modeled as $\Delta t(1-L(t))$. Therefore, the expected net revenue function can be developed from Eq. (5.1) with the consideration of system efficiency loss as:

$$R_{net} = \gamma \sum_{k=0}^{n_T} (1 - L(k\Delta t))\Delta t - C_{to} \tag{5.13}$$

where $n_T = T/\Delta t$ is the number of the discretization intervals.

5.4 Maintenance decision-making based on Monto-Carlo simulation

In this section, a prognostics-induced maintenance optimization framework is proposed, as shown in Fig. 5.2. MCS and GA are used to find optimized maintenance decisions. The considered system has N sub-systems, and each sub-system has n_j components connected in series. It is assumed that the lifetime distributions of all components are known in advance, and the parameter set Ω related to the efficiency loss has been estimated for the system. In addition, a database of prognostic errors D_e is generated in advance for all possible component lifetimes within a specific range of RMSEs, according to Section 5.2.2. Given components' lifetime distributions, efficiency model parameters Ω, database D_e, cost parameters C_{fs}, C_{ps}, and $C_{r,i}$, $i = 1,\ldots, n$, the number of planned inspections n_m, maintenance decision variables p_{pm}, p_{om}, and $t_{m,i}$, $i = 1,\ldots, n_m$, and revenue reward parameter γ, the following steps are summarized to estimate the expected net revenue for one simulation.

Step 1: Initialization. All parameters are specified. The total maintenance cost C_{to} and the baseline working time T_b are set to 0 at the beginning and are updated during the simulation process. The failure time, $T_{F,i,j}$, is defined for component $c_{i,j}$. Initially, The failure time $T_{F,i,j}$ is equal to the lifetime $l_{i,j}$ of the component sampled from its lifetime distribution. Based on the sampled lifetime and efficiency model parameters Ω, the system efficiency loss due to the degradation of each component is randomly generated by Eq. (5.6), and the total system efficiency loss can be obtained by Eq. (5.9). In addition, if the component $c_{i,j}$ can be monitored in real-time, a prognostic error curve is randomly sampled from D_e based on its lifetime, and the corresponding predicted RUL percentage is calculated.

Step 2: Calculate the system baseline working time. The time interval Δt is preset before the simulation. The planned operating period T is discretized into n_T time intervals based on Δt. At time $t = k\Delta t$, $k = 1, 2, \ldots, n_T$, the system baseline working time is calculated as $(1 - L(k\Delta t))\Delta t$. The baseline working time T_b is updated by $T_b = T_b + (1 - L(k\Delta t))\Delta t$.

Step 3: Perform planned inspection. At time t, if the inspection time is reached, a planned inspection is performed, and a replacement is decided based on the health status of each component as described in Section 5.2.3. The maintenance cost $C_{m,t}$ at time t is calculated based on Section 5.2.3, and the total maintenance cost C_{to} is updated by $C_{to} = C_{to} + C_{m,t}$. If the component $c_{i,j}$ is replaced, a new lifetime $l_{i,j}$, and the corresponding failure time $T_{F,i,j}$ is updated to $t + l_{i,j}$. At the

same time, a new prognostic error curve and system efficiency loss due to degradation of component $c_{i,j}$ is re-generated, and the total system efficiency loss and the predicted RUL percentage are recalculated. When a new lifetime $l_{i,j}$ is re-generated, imperfect maintenance can be considered by introducing a reduction factor. Details can be found in Section 5.5.3.

Step 4: Replace the failed component. At time t, if one component fails, i.e., $\min(T_F) \leq t$, corrective replacement is executed. Determine if any other components need to be opportunistically replaced, and calculate the maintenance cost $C_{m,t}$ at time t based on Section 5.2.3. The total maintenance cost C_{to} is updated by $C_{to} = C_{to} + C_{m,t}$. If the component $c_{i,j}$ is replaced, a new lifetime $l_{i,j}$ is regenerated, and the corresponding efficiency loss and prognostic error curve are re-generated.

Step 5: Perform preventive replacement. At time t, preventive replacement is executed if any components can be monitored and meet the replacement requirement described in Section 5.2.3. While conducting preventive replacement, determine if opportunistic replacement is required for other components and calculate the maintenance cost $C_{m,t}$. The total maintenance cost C_{to} is updated by $C_{to} = C_{to} + C_{m,t}$. If the component $c_{i,j}$ is replaced, similar to steps 3 and 4, a new lifetime $l_{i,j}$, and the corresponding efficiency loss and prognostic error curve are re-generated. If the component $c_{i,j}$ is not replaced, its failure time $T_{F,i,j}$ and other parameters are not updated.

Step 6: If the time reaches the planned period T, the simulation process is stopped. Otherwise, Steps 2, 3, 4, and 5 are repeated.

Step 7: When the simulation is stopped, the expected net revenue can be calculated based on the baseline working time T_b and the revenue reward parameter γ by Eq. (5.13).

Based on the idea of MCS, many times simulations are conducted in parallel, and the average value of the net revenue is calculated for the optimization process. The aim of this work is to find optimal maintenance decision variables $\{t_{m,i}$ $(i = 1,\ldots, n_m)$, p_{pm}, $p_{om}\}$ by maximizing the expected net revenue during the planned period T. In practice, the cost of sale can be considered in the objective function if the asset is sold after T. In addition, if the asset continues to be used in the next period, the final state of the asset in the first period can be set to the initial state of the next period, then we can use the same MCS procedure to find the optimal policies for the next period. Given a

specific number n_m, the corresponding optimal maintenance decisions $\{t_{m,i}\ (i = 1,\ldots, n_m), p_{pm}, p_{om}\}$ can be found according to the process described above by using GA. Since longer preventive maintenance intervals are usually scheduled in the early periods and shorter intervals in the later periods, we constrain t_m in the calculation by $t_{m,1} \geq t_{m,2} \geq \cdots \geq t_{m,n_m}$.

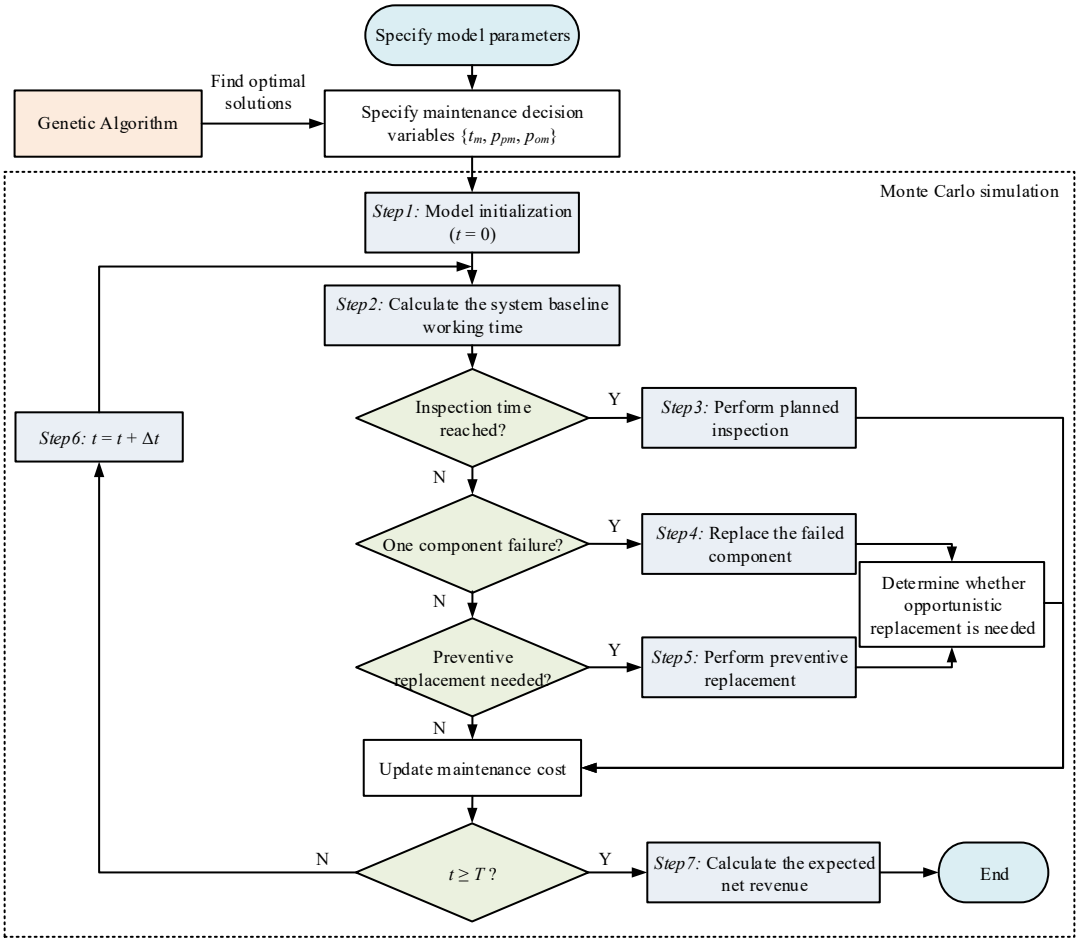


Fig. 5.2: Flow diagram of Monte Carlo-based maintenance optimization

5.5 Case study

In this section, a case of maintenance optimization on wind turbine farms is provided to demonstrate and validate the proposed method. In Section 5.5.1, the details of the wind turbine system and the data used are introduced. Then, in Section 5.5.2, comparative studies are conducted with time-based policies, as well as with the CBM policy only minimizing the maintenance cost and the policy maximizing expected net revenue without considering degraded working efficiency.

Finally, the influence of important parameters on the maintenance model, particularly the accuracy of the prognostic model, is discussed in Section 5.5.3.

5.4.1 System description

A case study from an onshore wind farm in the UK is considered in this work. The wind farm contains 13 wind turbines. The turbine type is GE Energy 1.5s with a rated power of 1500 KW, and the total power of the wind farm is 19500 KW [143]. A typical configuration of a wind turbine is shown in Fig. 5.3 [144]. Wind turbine blades collect the energy from the wind and then transmit the rotational energy to the gearbox. The gearbox further transfers the mechanical energy to the generator and ultimately generates electricity. All modern wind turbines are mounted with spherical roller bearings as main bearings to reduce the frictional resistance caused by the relative rotation of components. In this case, four key components in each wind turbine are studied, including the generator, bearing, gearbox, and rotor-blade. As essential parts of wind turbines, these components experience relatively high failure risks among all components [139].

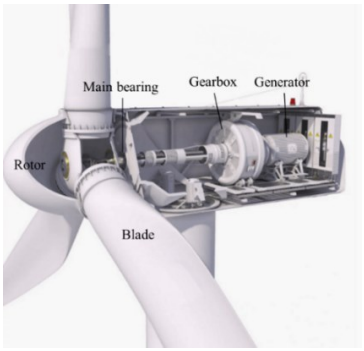


Fig. 5.3: Typical configurations of a wind turbine [144]

The degradation of bearings and gears is usually accompanied by increased vibration amplitude. In some wind turbines, the RULs of gearboxes and bearings can be predicted in real time based on accelerometer measurements. However, the damage or crack of the rotor-blade can only be detected by non-destructive testing (NDT) techniques during periodic inspections. Therefore, we consider the bearing and gearbox as the first type of components that can be continuously monitored, while the generator and rotor-blade are regarded as the second type of components that can only be detected during inspection. Commonly, the lifetime distribution of components can be selected based on failure time or suspended time data using the Akaike information criterion or Bayesian information criterion [145]. In this work, we assume that the time-to-failure distribution

of components is known in advance and follows the Weibull distribution [10][74]. The scale and shape parameters of Weibull distributions are placed in Table 5.1 based on the data in [74]. In addition, cost values and revenue parameters are set in Table 5.1 according to reported works [74][146] and widely used contracts [147], but with proper modifications.

Table 5.1: Failure parameters and maintenance cost data (time unit: month, cost unit: $k)

Components	Distribution parameters		Maintenance costs			Revenue model parameters
	Scale	Shape	C_r	C_{ps}	C_{fs}	γ
Generator	110	2	100	10 per wind turbine	100 per wind turbine	350
Bearing	125	2	60			
Gearbox	80	3	150			
Rotor-blade	100	3	120			

In the simulation, we set the total period T to 25 years, as wind farms typically have a life cycle of around 25 years [148]. Wind turbines will be renewed after the planned time T. We assume a repair will take 0.5 months if a component needs to be replaced. Otherwise, a preventive inspection will take approximately 0.2 months. In addition, when multiple wind turbines are being maintained at the same time, the setup cost of the extra wind turbine is cut in half. For example, if corrective replacement and preventive replacement are performed on λ_1 and λ_2 wind turbines, respectively, the setup cost will be $C_{fs} + 0.5(\lambda_1 - 1)C_{fs} + 0.5\lambda_2 C_{ps}$. If only preventive replacement is performed on λ ($\lambda > 1$) wind turbines, the setup cost will be $C_{ps} + 0.5(\lambda - 1)C_{ps}$.

To estimate the parameter set Ω of system efficiency loss, a total of 1687 farm-year load factor data [5] covering onshore turbines built in the UK from 1991 to 2012 is collected. Fig. 5.4 shows the statistical load factor histories of 168 wind farms with more than five years of data, using a 12-month moving average to smooth out seasonal variations. Oftentimes, the system load factor tends to rise in the first year of operation as turbines are still being commissioned [5]. We therefore start our parameter estimation with data from the second year of each farm. The system efficiency loss is calculated from load factor histories by repeated sampling. The effect of structure dependencies on system efficiency loss is ignored in this work. The weights of the impact of each component degradation on the efficiency loss of wind turbine are referred to reported works [149][150], where the weights of the generator, bearing, gearbox, and rotor-blade are set to 0.2, 0.2, 0.3, and 0.3.

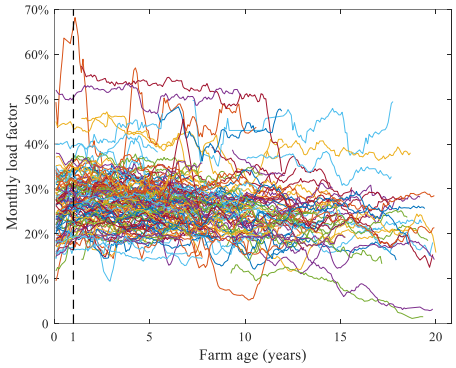


Fig. 5.4: Decline profile of load factor for UK farms aggregated by age

5.4.2 Comparative results

We start discussing the benefits of the proposed maintenance policy by comparative studies with an initial set of parameters, i.e., the base case. Before the simulation, the efficiency model parameters are learned from the load factor data shown in Fig. 5.4, and we get $\Omega = \{\theta_1, a_1, a_2, a_3, a_4, \sigma_1, \sigma_2, \sigma_3, \sigma_4\} = \{5.3238, 1.3108, 1.7447, 1.8714, 1.8186, 13.5047, 17.4193, 13.4691, 19.2618\}$. Based on the simulation process proposed in Section 5.4, the optimal maintenance decisions can be obtained given a specific number n_m of planned inspection intervals within the planned period T. The prognostic model is assumed to be perfect in the base case, i.e., its RMSE equals 0. The influence of the prognostic model on the performance of CBM will be discussed in the next section. In the simulation, the time interval Δt is set to 1 month. The RULs of the bearing and gearbox can be obtained by the prognostic model at each time interval. It is assumed that component replacement of wind turbines takes 0.5 months, while preventive maintenance without replacement takes only 0.2 months. The optimization results of the proposed policy are presented in the last row of Table 5.2. If the sum of the planned inspection intervals is less than the planned period T, the subsequent inspection intervals will keep the last optimized time. In other words, we obtain the planned inspection intervals of the proposed policy as {47, 37, 35, 32, 29, 29, 19, 14, 12}, which means that the initial eight inspection intervals are 47, 37, 35, 32, 29, 29, 19, 14 months, and all subsequent inspection intervals are 12 months.

To validate the superiority of the proposed maintenance optimization method, five maintenance methods are simulated simultaneously for comparison. These strategies aim to find optimal solutions by minimizing the maintenance cost, including time-to-failure maintenance, i.e., corrective maintenance (CM), age-based maintenance (AM), constant interval maintenance (CI),

maintenance with varying intervals (VI), and prognostics-induced CBM (PCBM). The policies of AM, CI, and VI are TBM policies, which perform maintenance actions with constant scheduled time intervals, see [151]. When maintenance or inspection is performed, the component will be replaced if it reaches the last in-service health level, i.e., RUL percentage from 0 to 0.05. PCBM is a CBM policy that considers corrective, preventive, and opportunistic replacements, as shown in Section 5.2.3. MCS and GA are also used to find the optimal maintenance decisions from these maintenance policies. The comparative results of different maintenance policies are listed in Table 5.2. Overall, the optimized maintenance decisions of both TBM and CBM policies require lower maintenance costs than CM, and the proposed method outperforms the other strategies because it has the highest expected net revenue. AM, CI, VI, PCBM, and the proposed method achieve 14.0%, 16.0%, 18.0%, 22.2%, and 21.0% maintenance cost reduction and 2.4%, 2.1%, 4.2%, 9.0% and 10.1% revenue improvement compared with CM, respectively. Since preventive maintenance will inevitably reduce system availability, even though the optimized TBM decisions can reduce maintenance costs, the improvement in system net revenue is not significant. In contrast, the expected net revenue can be greatly improved when the prognostic information is induced into the maintenance actions. Although the proposed CBM optimization method finds maintenance decisions with higher maintenance costs compared to PCBM, its maintenance decision has longer inspection intervals and thus improves the system availability and increases expected net revenue.

Table 5.2: Comparison of different maintenance strategies (time unit: month, cost unit: $k)

Strategies	Maintenance decision variables	Maintenance cost	Expected net revenue ($k/year)
CM	–	1220.1669	2509.0045
AM	$t_m = 4$	1049.0201	2568.4151
CI	$t_m = 4$	1024.6566	2562.8292
VI	$t_m = \{22, 18, 9, 8, 4\}$	1000.2837	2615.0426
PCBM	$t_m = \{44, 29, 28, 17, 17, 12, 10, 7, 6\}$, $p_{pm} = 0.0273$, $p_{om} = 0.0490$	948.8787	2735.4389
Proposed method	$t_m = \{47, 37, 35, 32, 29, 29, 19, 14, 12\}$, $p_{pm} = 0.0392$, $p_{om} = 0.0667$	964.5079	2763.3745

5.4.3 Sensitivity analysis

Although prognostics-induced CBM allows for significantly effective maintenance actions, its relative performance strongly depends on the maintenance cost structure, the production margin, the performance of each maintenance, and the accuracy of the prognostic model. Therefore, the above factors are considered in this section in conjunction with the proposed prognostics-induced CBM policy. The CI maintenance policy has been widely used in actual industries [14][151]. Therefore, in the sensitivity analysis, the optimized net revenue undertaken by the proposed method is compared with the policies of CM and CI. We set the expected net revenue obtained by CM policy as the baseline revenue, i.e., 0. The sensitivity results show the relative net revenue of other policies compared to the CM policy.

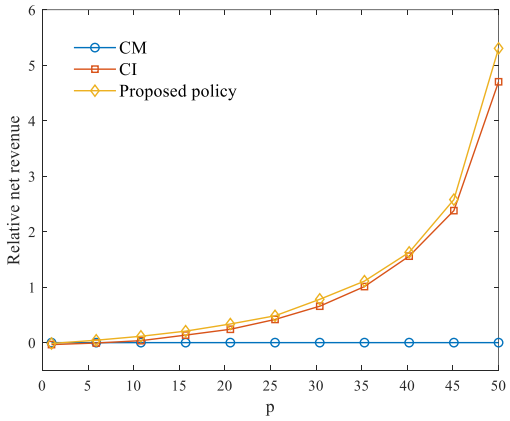


Fig. 5.5: Relative net revenue under optimal maintenance policies with different cost structures

We first assess the effect of changing the behavior of the cost structure on the expected net revenue. We use a proportional parameter $p = C_{fs} / C_{ps}$ as a measure of this variation. The value of p will always be larger than 1, and p is equal to 10 under the base case. We adjust p from 1 to 50 by changing the value of C_{fs}, and the corresponding relative net revenues are compared in Fig. 5.5. It can be observed that the relative net revenue of CI and the proposed policy shows exponential growth as the value of p increases. However, the increase in the gap of the relative net revenue between CI and the proposed policy is not significant. This is mainly because the net revenue from the CM policy decreases rapidly as the value of C_{fs} increases, while in comparison, the expected net revenues optimized by CI and the proposed policies are relatively stable. If the failure setup cost C_{fs} is extremely small, preventive maintenance will lead to a reduction in system availability,

and thus, the relative net revenue of different maintenance policies will be slightly less than zero. In contrast, if the ratio of C_{fs} and C_{ps} is extremely large, preventive maintenance can be performed very frequently to improve the system reliability, so the expected net revenues from using maintenance policies will not sharply decrease.

We continue to explore the effect of the production margin on the performance of maintenance policies. The value of reward rate γ in the net revenue model is changed from -50% to 50% in the sensitivity analysis, and the expected net revenues obtained by different policies are compared, shown in Fig. 5.6. It is worth mentioning that the maintenance actions optimized by all these policies will not change with different reward rates γ, which means that their maintenance cost will not change. It can be observed from Fig. 5.6 that the relative net revenue of all policies relative to the CM policy flattens out as the γ increases. With different γ, the proposed policy can always find the best decision with the highest net revenue. When the reward rate γ is extremely large, the reduction of system availability caused by PM actions will result in more revenue reduction. Therefore, the expected revenue optimized by the CI maintenance is smaller than that of the CM policy when the change in reward rate is greater than +40%.

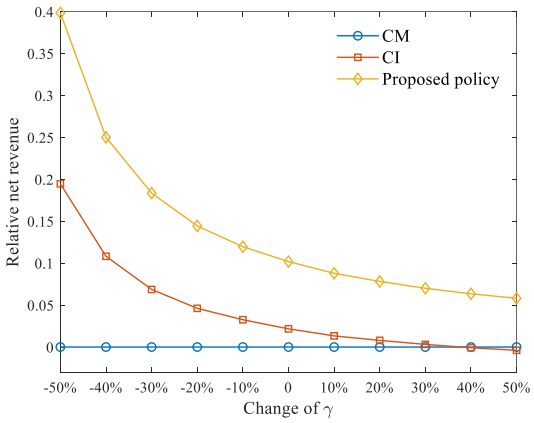


Fig. 5.6: Relative net revenue under optimal maintenance policies with different production margins

As mentioned in Step 3 in Section 5.4, imperfect maintenance can be considered by introducing a reduction factor. We therefore consider a ratio reduction factor from 0 to 1 in the MCS, denoted by b. When a new lifetime $l_{i,j}$ of component $c_{i,j}$ is sampled, its lifetime will be corrected to $bl_{i,j}$. Fig. 5.7 shows the comparative net revenue optimized by different policies with the variation of b from 0.5 to 1. As shown in Fig. 5.7, compared to CM, the proposed policy can perform well, especially

in the case of imperfect maintenance performance, while the performance improvement of the CI policy is not significant for different reduction factors. Under the CI policy, relative net revenue increases slightly when b increases from 0.7 to 0.75, as its optimization interval increases from 3 to 4 months, leading to higher system availability. When b is small, i.e., the maintenance is imperfect, more component failures occur during the planned period, and therefore, significantly less revenue can be obtained under the CM policy. In contrast, the proposed policy receives much less impact from different b than CI and CM policies due to the use of prognostic information.

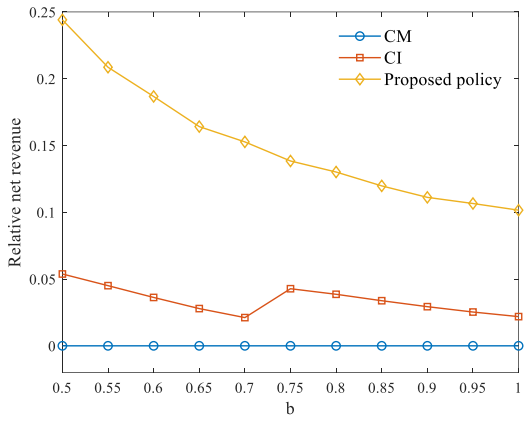


Fig. 5.7: Relative net revenue under optimal maintenance policies considering imperfect maintenance

Finally, it is the first time to discuss the influence of prognostic model accuracy on the benefit of CBM. Recalling the mention in Section 5.2.2, the RMSE of a prognostic model often varies from 0.1 to 0.2 in mature applications. Therefore, we construct a database D_e containing errors with the value of RMSE from 0 to 0.3 using the method provided in Section 5.2.2 with the random initial values from 0 to 0.5, model parameters θ_e from 1 to 5, and σ_e from 0 to 2. Fig. 5.8 presents three simulated prognostic profiles with different RMSEs as examples. With the proposed method, the simulated RUL prediction errors can always converge to a value close to zero over time. When the prognostic error is introduced, only the performance of the proposed policy considering the prognostic information fluctuates with changes in RMSE value. In the sensitivity test, the optimal maintenance decision of the proposed policy shown in Table 5.2 will not change, and the calculated relative net revenues for various values of RMSE are shown in Fig. 5.9. In Fig. 5.9, the x-axis coordinate RMSE of 0.02 corresponds to the simulated RMSE from 0 to 0.02, and so on. Obviously, the relative revenue curve shows an inflection point at RMSE = 0.12. When the RMSE of the

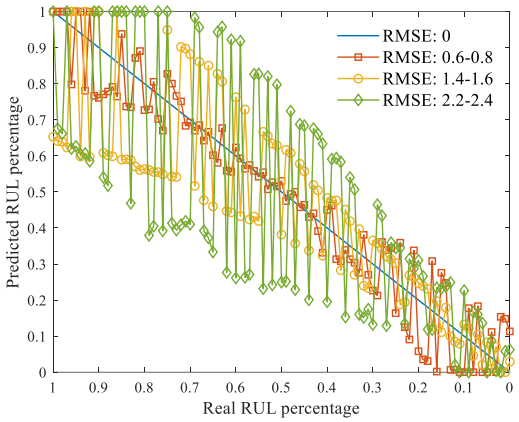


Fig. 5.8: Simulated prognostic profiles with different RMSEs

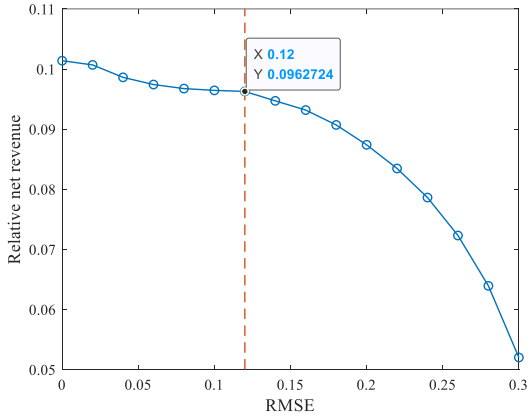


Fig. 5.9: Relative net revenue of the proposed policy considering prognostic errors

prognostic model is less than 0.12, the revenue curve is flatter relative to the RMSE value. In this region, even a 50% improvement in the performance of the prognostic model, i.e., RMSE from 0.12 to 0.06, will result in only a 1.2% improvement in the relative net revenue. In contrast, when the RMSE of the prognostic model is greater than 0.12, the relative revenue shows an exponential decrease with increasing RMSE. When we increase the RMSE of the prognostic model from 0.2 to 0.16, i.e., a 20% improvement, the maintenance policy can obtain a 6.6% benefit improvement. And when the RMSE of the prognosis model is increased from 0.3 to 0.12, i.e., a 60% improvement, the benefit improvement can be as high as 75%. It is worth mentioning that obtaining a prognostic model with an RMSE of about 0.2 is usually not difficult, even with a small amount of historical data, as discussed in Chapter 2. However, it is very challenging and consuming to obtain sufficiently accurate RUL prediction models with RMSE smaller than 0.1, which requires a large amount of training data and time. Therefore, it is necessary to analyze the impact of the prognostic

error on maintenance behavior before using the CBM policy to find the best maintenance decision with limited spending.

5.6 Conclusions

In this chapter, a prognostics-induced CBM optimization method is proposed to find optimal maintenance decisions with maximum net revenue. Different from reported works, this work considers that only some parts of the system can be monitored continuously and optimizes both planned inspection intervals and prognostic thresholds, i.e., preventive and opportunistic thresholds, in maintenance decision-making. A data-driven efficiency modeling method is proposed and incorporated into the net revenue optimization framework to consider the economic loss due to the system degradation when assessing the maintenance benefits. Based on the constructed net revenue, MCS and GA are used to find optimal maintenance decisions. The findings of the comparative case study validate the performance and usability of the proposed method.

We further extend our model in the sensitivity analysis to analyze the effects of several practical factors on the benefit of CBM. In particular, this work is the first to discuss how the accuracy of prognostic models affects the economic benefits of CBM policy in complex multi-component systems. The proportional difference between CM and preventive maintenance setup costs turns out to be more important for the relative revenue benefit of CBM than other factors. Even though changes in some factors can lead to large variations in CBM benefits, the CBM induced by prognostics can always perform significantly better than other maintenance policies, such as time-based policies. However, this work ignores the effect of structure dependencies on system efficiency loss and is limited by not considering the improvement in component RUL from non-replacement preventive maintenance. In addition, when repairs are required, the proposed approach assumes that the maintenance can always start immediately. These limitations can be considered in future work.

Chapter 6: Condition-based maintenance for wind farms with imperfect prognostic information

When the prognostic information is not accurate and comprehensive enough, it is unreliable to determine maintenance actions based only on predicted remaining useful life. This chapter builds upon the work in Chapter 5 by developing focused maintenance optimization strategies for a more realistic system, i.e., wind farm, involving condition monitoring with auxiliary inspection information. A new maintenance basis is proposed to calibrate prognostics with the help of rough estimates of the turbine component lifetime from inspections. The materials in this chapter are covered by the fifth research topic (Topic #5), which is introduced in Section 1.3. The organization of this chapter is as follows. In Section 6.1, research and challenges in scheduling maintenance for wind farms are presented. Section 6.2 introduces the proposed maintenance basis expressed as posterior RUL percentage. In Section 6.3, the maintenance costs and the optimization model are formulated, and the simulation procedure of maintenance actions is illustrated. Section 6.4 presents the results and discussions through a numerical example of wind farm maintenance optimization. Finally, conclusions are made in Section 6.5. The results of this chapter are submitted as a journal paper.

6.1 Introduction

Recent years have witnessed a rapid growth of renewable energy industries because of their sustainability and environmental friendliness in facing the energy and climate challenges [152]. Wind power is an essential type of renewable energy that has contributed to over 20% of total generated electricity in countries like Germany, Denmark, and the United Kingdom [153]. Operation and maintenance costs account for approximately 14–30% of the total expenditure on wind farm generation, making it one of the most substantial cost components in turbine assets [154][155]. Maintenance actions and the downtime loss caused by turbine failures are responsible for around 60% of the overall maintenance costs [153].

Condition-based maintenance (CBM) enables the most efficient scheduling of maintenance based on sensor data from turbine components, ideally leading to maintenance being performed just

before the component failure. However, the evaluation of condition data is restricted by the actual operating environment of wind turbines. This limitation can lead to higher errors in predictions, ultimately causing a rapid progression of CBM benefits in the negative direction [14]. In the trend of demanding higher production efficiency, wind turbines tend to be installed with larger power capacity and rotor sizes and operate in harsher environments, e.g., in deep waters far from shore [156]. However, such development strategies may be accompanied by higher failure rates, resulting in potentially up to four times higher maintenance costs [157][158]. Wind turbines may present a hub height of up to 150 m and a rotor diameter of up to 160 m and are often geographically distributed in a wind farm [159]. Maintenance teams face challenges in accessing these turbines due to their large scales and harsh environments, which require earlier preparation and even the use of costly transportation, such as helicopters [92]. Moreover, the measurements of some turbine components, e.g., rotor-blade strain, not only undergo significant fluctuations due to environmental variations but are also challenging to gather comprehensively [160], making accurate monitoring of all critical components unfeasible. Despite the potential of CBM to increase the profitability of wind farms using condition data, these practical factors make the CBM of wind farms a very challenging task.

The works that combine condition monitoring data and opportunistic maintenance at the wind farm level are not abundant. One of the initial studies [161] optimized two predictive thresholds for wind farm maintenance using prognostics. In this work, prognostic uncertainty was modeled as a normal distribution with a mean of zero, so no prediction bias was taken into account. Zhou and Yin [92] built upon the work in [161] by enabling the maintenance decisions to adapt to cases with long preparation times. As proposed by [92], the numerical simulation method is flexible for maintenance optimization to consider strong dependencies and opportunities. However, this study only provided a basic description of economic dependencies at the wind farm level and was still limited to zero-mean error prognostics. In [153], a more realistic wind farm model was created, where the economic dependencies within a turbine and within a farm were described in terms of crane cost and transportation cost, respectively. Bakir et al. [162] determined maintenance opportunities that exhibit variations across varying wind turbine conditions. In this research, the RUL distribution was predicted by stochastic degradation models, which always provide valid prognostic information while accounting for realistic uncertainty.

It may be noted that most CBM strategies proposed for wind farms are not fundamentally different, regardless of whether the wind farm operates in a remote or harsh environment [163]. Previous studies that incorporated prognostics into wind farm maintenance either considered the predicted RUL as an average value [164] or as a distribution [92][153][162]. However, these studies were limited by the availability of accurate and comprehensive prognostic models based on historical data. If the prognostic information is not sufficient, making maintenance decisions based on prognostics alone may cause over-maintenance or under-maintenance of turbine components [92]. To the best of our knowledge, there is no CBM policy that specifically considers the maintenance of wind farms working under harsh conditions. These challenges are highlighted not only by high setup costs and strong economic dependencies, but more significantly by partial monitorability and potentially inaccurate prognostic information. This forms the contribution of this work.

As inspection techniques indirectly measure the health of turbine components, they can provide valuable information complementary to prognostics in reducing the impact of prediction bias on maintenance actions. This chapter explores the potential of combining additional inspection information to deal with CBM in the case of imperfect prognostics. The maintenance costs and opportunistic maintenance policies are modeled to capture the inter-dependencies between components within a turbine, as well as components across turbines in the farm. An optimization method that integrates component-level prognostics and inspection estimates into a large-scale numerical model is proposed, particularly for wind farms operating in challenging conditions.

The main contribution lies in dealing with the issue that the wind turbine is partially monitorable and RUL predictions are not accurate enough during maintenance optimization. The proposed optimization approach identifies maintenance thresholds while also optimizing the planned inspection intervals at the wind farm level. A new maintenance basis denoted by the posterior RUL percentage is presented to determine preventive maintenance actions for wind turbines. This maintenance basis integrates auxiliary health estimates gathered during inspections with predictive analytics to calibrate RUL predictions, allowing for better decision-making to enhance maintenance benefits, especially in situations where prognostics is imperfect.

6.2 Maintenance basis allowing for inaccurate prognostics

Various models are available from physics-based approaches to data-driven methods for RUL predictions of wind turbine components [165]. The predicted RUL can be presented as a distribution consisting of RUL percentages from 0 to 1 with corresponding probabilities. The failure history of a turbine component is the condition monitoring data, e.g., oil debris or vibration data, collected from the beginning to the end of the component's life. However, collecting a sufficient number of failure histories is often challenging, resulting in biased RUL predictions due to limited failure histories that cannot cover all possible conditions of wind turbines operating in a real-world environment. In addition, it is impractical to install sensors for all turbine components to monitor their health status in real time. Hence, relying only on component prognostics is inadequate to develop a standard maintenance threshold for wind turbines that can help determine maintenance plans.

The prognostics illustrated in Fig. 6.1 present two common situations that exhibit notable biases for maintenance scheduling, where the predicted RUL percentage indicates the mean value simulated by the stochastic method described in Section 3.2. When the prognostic model lacks precision, the predicted RUL may fall to 0 well before the actual end of life or remain significantly high despite the component failure, possibly due to inaccurate estimates of the degradation threshold in the prognostic model. Therefore, setting a higher maintenance threshold can reduce the occurrence of component failures in case #2 but, conversely, result in premature maintenance operations and economic waste in case #1.

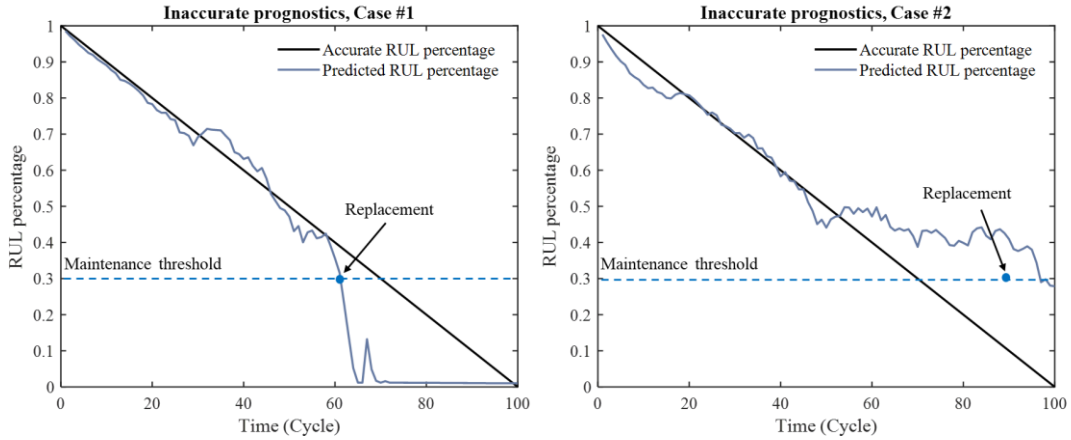


Fig. 6.1: Two typical cases of inaccurate prognostics for maintenance scheduling

To overcome improper maintenance planning due to partial and erroneous prognostics, we optimize predictive maintenance thresholds while also finding planned inspection intervals at the wind farm level. It is assumed that the health of a turbine component can be roughly estimated during the inspection when it is in a sub-health state. For instance, as blade cracks grow to a specific length, it is reasonable for maintenance teams to make a range estimate of blade lifespan based on detected cracks, which also applies to rough estimates of bearing and gearbox health in terms of vibration amplitude. In this work, we assign three degradation levels for a turbine component, i.e., I status, II status, and the last in-service status, corresponding to a range of remaining lifetimes. Maintenance procedures are primarily governed by the predicted RUL when prognostics is deemed trustworthy. On the contrary, the objective of prognostics is to prevent undesired turbine component failures between inspection intervals as much as possible. Once the inspected turbine component is in the last in-service status, the component will be replaced [166]. If the detected component is in I or II status, the corresponding rough life estimate can be utilized to adjust prognostic results. To this aim, a new maintenance basis denoted by the posterior RUL percentage is proposed to incorporate inspection information into RUL predictions for better maintenance plans.

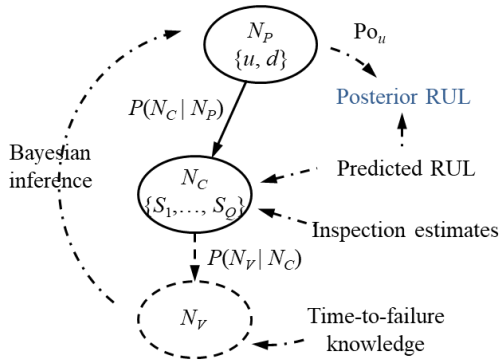


Fig. 6.2: Diagram of Bayesian network for posterior RUL estimation

The failure time $\hat{t}_f$ of a turbine component can be estimated with prognostic information in real-time based on its average predicted RUL percentage $\bar{\hat{r}}$ and age t by $\hat{t}_f = t/\left(1-\bar{\hat{r}}\right)$. When a turbine component is inspected to be in I or II status, the approximate range of its failure time can be determined in terms of upper limit time $t_{f,u}$ and lower limit time $t_{f,l}$. The posterior RUL percentage is derived by a two-node Bayesian network, as shown in Fig. 6.2. In the Bayesian network, the parent node N_P has two functional states, i.e., upward (u) and downward (d). The probability of

states u and d indicate the proportion of the predicted failure time $\hat{t}_f$ moving toward the upper and lower limit time, respectively. Initially, the prior probability of node N_P is set to 0.5.

The state of the child node N_C is divided into Q discretization states $S = \{S_1, S_2, \ldots, S_Q\}$, where Q should be an even number, recording possible failure time points. The time points recorded in the states from S_1 to $S_{Q/2}$ are shorter than the predicted failure time, in contrast to the records in the states from $S_{Q/2+1}$ to S_Q. The maximum and minimum times corresponding to states S_1 and S_Q are determined dynamically by the relationship among $\hat{t}_f$, $t_{f,u}$, and $t_{f,l}$. As an example, if $\hat{t}_f$ is close to $t_{f,l}$, the maximum and minimum times will be set to $t_{f,u}$ and $2\hat{t}_f - t_{f,u}$. Conversely, if $\hat{t}_f$ is closer to $t_{f,u}$, these two times will be defined as $2\hat{t}_f - t_{f,l}$ and $t_{f,l}$. The time points between states S_1 and S_Q are separated equally. Given that the state of the parent node is upward (u) or downward (d), the conditional probability that the child node state is S_q, $q = 1,\ldots, Q$, can be defined by:

$$P\left(N_C = S_q \mid N_P = u\right) = \frac{\int_{(q-1)/Q}^{q/Q} f_u(x)dx}{\int_0^1 f_u(x)dx} \tag{6.1}$$

$$P\left(N_C = S_q \mid N_P = d\right) = \frac{\int_{(q-1)/Q}^{q/Q} f_d(x)dx}{\int_0^1 f_d(x)dx} \tag{6.2}$$

where $f_u(x)$ is the probability distribution function of a normal distribution with mean one and standard deviation σ_u, and $f_d(x)$ is the probability distribution function of a normal distribution with mean zero and standard deviation σ_d. The standard deviations σ_u and σ_d are determined to measure the movement of $\hat{t}_f$ to $t_{f,u}$ and $t_{f,l}$, respectively. It is noted that in this approach, only two standard deviations need to be adjusted manually. Therefore, the performance of RUL calibration is limited to the proper configuration of these two parameters.

A virtual evidence node N_V [167] connected with the child node is added to assess the possibility of failure time points corresponding to each state S_q. If the time point of state S_q is between $t_{f,l}$ and $t_{f,u}$, the conditional probability $P\left(N_V \mid N_C = S_q\right)$ is the time-to-failure probability at the current moment. Otherwise, the conditional probability is 0. The posterior upward probability is represented as follows:

$$\begin{aligned} \mathrm{Po}_u &= P(N_P = u \mid N_V) \\ &= \frac{P(N_V \mid N_P = u)}{P(N_V \mid N_P = u) + P(N_V \mid N_P = d)} \end{aligned} \tag{6.3}$$

where $\mathrm{Po}_u > 0.5$ indicates that the predicted RUL may fall to 0 early and need to be adjusted upward, while $\mathrm{Po}_u < 0.5$ indicates that even if the turbine component is close to failure, its predicted RUL is still high and needs to be adjusted downward. Therefore, the predicted failure time can be updated as in Eq. (6.4), and the posterior RUL percentage can be estimated by Eq. (6.5).

$$\hat{t}_{f,\mathrm{Po}} = \begin{cases} \left(\mathrm{Po}_u - 0.5\right)\left(t_{f,u} - t_{f,l}\right)/0.5 + t_{f,u}, & \text{if } \mathrm{Po}_u \le 0.5 \\ \left(\mathrm{Po}_u - 0.5\right)\left(t_{f,u} - t_{f,l}\right)/0.5 + t_{l,u}, & \text{if } \mathrm{Po}_u > 0.5 \end{cases} \tag{6.4}$$

$$\hat{r}_{\mathrm{Po}} = \frac{\hat{t}_{f,\mathrm{Po}} - t}{\hat{t}_{f,\mathrm{Po}}} \tag{6.5}$$

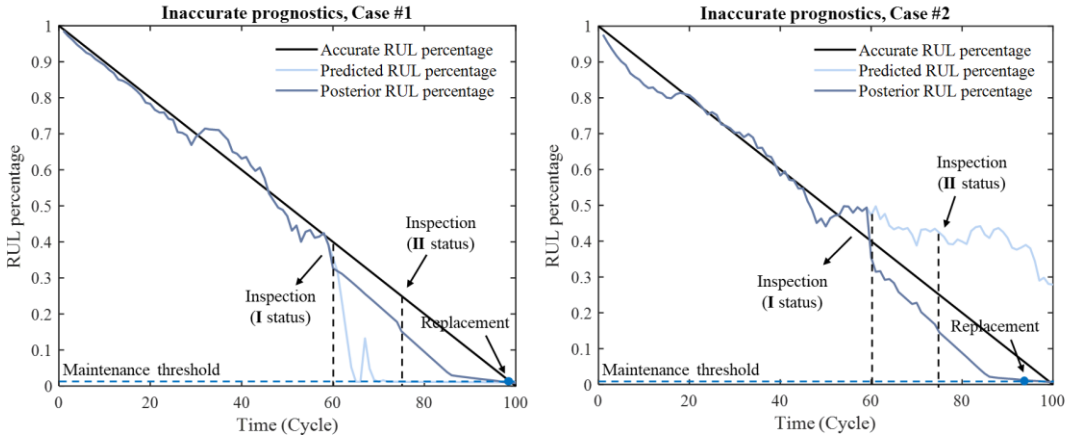


Fig. 6.3: Posterior prognostic information for maintenance scheduling

The posterior RUL percentage quantifies the trade-off between prognostics and inspections. As an illustration, Fig. 6.3 shows the capability of the proposed posterior RUL for maintenance deployments in two cases shown in Fig. 6.1 with inaccurate prognostics, where the inspection interval is 15 cycles. The posterior RUL accurately corresponds to the predicted RUL before the component life can be roughly estimated during the inspection. If the component is inspected to be in I or II status, the inspection information is incorporated to calibrate predictions, thus avoiding large biases in prognostics when the component is close to failure. This approach enables the optimization of a standard predictive maintenance threshold to overcome both over-maintenance

and under-maintenance of turbine components, particularly when the predicted RUL is not accurate enough.

6.3 The proposed CBM approach for wind farm

6.3.1 Formulation of the wind farm model

Consider a wind farm consisting of N wind turbines of the same type. Each wind turbine has M critical components. Let $c_{i,j}$ represents the i-th component in the j-th wind turbine, $i = 1, 2,\ldots, M$, and $j = 1, 2, \ldots, N$. The wind turbine is a series system, with the components connected in series. The failure of any component in a wind turbine will lead to wind turbine failure.

The components of the turbine are classified into two major categories based on whether or not their health status can be monitored online in real time by sensors. The prognostic models that rely on online measurements can predict the remaining useful life (RUL) of the first type of components, which are continuously monitored. However, for the second type of component, their health status can only be identified during planned inspections since no sensors can be placed to measure their conditions. For instance, accelerometers can be used to measure the vibration of bearings and gearboxes in some wind turbines, and their RULs can be predicted based on the measurements. The defect of the rotor-blade can only be detected manually by non-destructive testing techniques during planned periodic inspections. We define the health status corresponding to the RUL percentage from 0.3 to 0.5, 0.1 to 0.3, and 0 to 0.1 as the I status, II status, and the last in-service status, respectively. The turbine status is assumed to be accurately identified during inspections or maintenance.

6.3.2 Simulation of predicted remaining useful life profiles

Turbine components deteriorate over time as a result of usage and aging. As these components degrade, the errors in predicting their RUL tend to approach zero [126]. However, most studies on maintenance optimization assume that the prognostic errors follow a normal distribution with a mean of zero and a constant standard deviation [92][153], which may not fully capture the actual prediction results. To model the predicted RULs of turbine components with realistic errors, the degradation of turbine components is defined as a generalized parametric form by $D(t) = \varphi(t;\mathrm{K},\Theta) + \varepsilon(t;\sigma)$, where $\varphi(t;\mathrm{K},\Theta)$ denotes the degradation function with deterministic

parameter sets K and stochastic parameter sets Θ, and $\varepsilon(t;\sigma)$ models the uncertainty with variance parameter σ. Specifically, $D(t)$ is formulated as an exponential model with Brownian motion error by Eq. (6.6), which has been widely used to characterize the degradation of bearings and gears [168][169]. In this formulation, ϕ is a constant deterministic parameter, $\Theta = [\theta, \beta]$ is the stochastic parameter sets with two parameters that follow normal distributions, and $\varepsilon(t)$ is a zero-mean Brownian motion error term with variance $\sigma^2 t$.

$$D(t) = \phi + \theta e^{\beta t + \varepsilon(t) - \frac{\sigma^2 t}{2}} \tag{6.6}$$

Given a predetermined failure threshold Λ, the failure time T_F, i.e., the time that the degradation signal first crosses the threshold, can be obtained. We use Eq. (6.6) to model a set of degradation signals given arbitrary parameters ϕ, Θ, and σ as the real degradation of turbine components, and the real RUL percentage at time t_k can be calculated by $r_k = (T_F - t_k)/T_F$. Meanwhile, given initial estimates for the distribution of stochastic parameters, its distribution can be updated to the posterior distribution, denoted by $\upsilon(\Theta)$, based on the real degradation signals up to the current time using Bayesian inference. Therefore, for each failure history generated by Eq. (6.6), the predicted failure time can be evaluated by $P\left(\hat{T}_F = \tau\right) = P\left(\tau = \min\left(t \geq 0 \mid D\left(t \mid \upsilon(\Theta)\right) \geq \Lambda\right)\right)$ as illustrated in [168][169]. Accordingly, the distribution profiles of the predicted RUL percentage $\hat{r}$ versus the real RUL percentage can be obtained. The root mean squared error (RMSE) is estimated based on the real RUL percentage r and the mean of the predicted RUL percentage $\bar{\hat{r}}$ to measure the prognostic errors as in Eq. (6.7).

$$RMSE = \sqrt{\Delta r \sum_{r=0}^{1} \left(r - \bar{\hat{r}}\right)^2} \tag{6.7}$$

Using this approach, it is feasible to generate a prognostic history of turbofan components. This history includes the actual RUL percentages ranging from 0 to 1, and their corresponding distributions of predicted RUL percentages. The predictions of RUL follow the model-based approach, ensuring that the predicted RUL converges to the true RUL over time. It is noted that $D(t)$ can also be formulated using other models, such as linear models. However, the choice of degradation models will not significantly influence the simulated prognostic profiles within a specific range of RMSE. During the simulation, multiple prognostic histories are generated in parallel, and the histories that satisfy the required RMSE range are saved in a database D_e. When

the database D_e grows larger, the nonlinear probability density at each RUL percentage can be directly sampled.

6.3.3 Opportunistic CBM policy for wind farm

A CBM policy is developed for the wind farm, consisting of corrective, preventive, and opportunistic replacements. Three decision variables are considered to control the maintenance frequency, including planned inspection interval t_m, preventive threshold p_{pm}, and opportunistic threshold p_{om}. Maintenance decisions are made for turbine component $c_{i,j}$ by its posterior RUL percentage calculated based on inspection and prognostic information, as well as by the planned inspection interval and maintenance opportunities with other components and turbines. The inspection interval and two predictive thresholds are determined to be the same for different turbine components to provide a uniform maintenance policy for the whole wind farm. The proposed CBM policy is summarized as follows:

- Perform failure replacement if a turbine component fails.
- If the inspection time is reached, perform planned inspections, and replace the turbine component if it reaches the last in-service health level with an RUL percentage less than 0.1.
- Perform preventive replacement on components in wind turbines if the component can be monitored in real-time and its posterior RUL percentage is less than p_{pm}.
- Perform opportunistic replacement for turbine component $c_{i,j}$ when a) any other component in the j-th wind turbine is determined to be maintained, and the component $c_{i,j}$ reaches the last in-service state with RUL percentage less than 0.1, and b) any other wind turbine fails or is preventively replaced, the turbine component $c_{i,j}$ can be monitored, and its posterior RUL percentage is less than p_{om}.

6.3.4 Maintenance cost modeling

Maintenance costs are defined to capture the economic dependence between the wind turbines and components. The variable cost of preventive replacement for the i-th turbine component, given in Eq. (6.8), consists of part cost $C_{R,i}$ and the corresponding setup cost $C_{P,i}^{S}$ defined as the cost required for labor operations and compensation of revenue loss due to downtime during the maintenance. Preventive and opportunistic replacements share the same maintenance cost.

$$C_{P,i} = C_{R,i} + C_{P,i}^{S} \tag{6.8}$$

The failure replacement cost for the *i*-th turbine component is defined as in Eq. (6.9), where $C_{F,i}^{S}$ is the setup cost of failure replacement for the *i*-th turbine component. Performing maintenance for a failed component often results in the same part cost but a higher setup cost than preventive replacement because of the potentially longer maintenance operations and downtime [153].

$$C_{F,i} = C_{R,i} + C_{F,i}^{S} \tag{6.9}$$

The planned inspection cost for the turbine component $c_{i,j}$ associated with the inspection operation and whether component maintenance is performed is expressed as in Eq. (6.10). In this formulation, $C_{I,i}^{S}$ signifies the setup cost incurred for the planned inspection, including expenditures associated with labor inspection operations, e.g., non-destructive testing, as well as revenue loss due to downtime, and other related costs. The inspection setup cost $C_{I,i}^{S}$ of a turbine component is often less than the preventive setup cost $C_{P,i}^{S}$, since the inspection can be done within a short period. $I_{R,i,j}$ is a binary variable indicating if the component $c_{i,j}$ needs to be replaced during the inspection. $I_{R,i,j}$ = 1 if the component is detected at the last in-service health level and is determined to be replaced, and $I_{R,i,j}$ = 0 otherwise.

$$C_{I,i,j} = I_{R,i,j} C_{P,i} + C_{I,i}^{S} \tag{6.10}$$

The determination of revenue loss in Eq. (6.8)-(6.10) involves multiplying the downtime by the power rate. While the power rate may vary depending on the wind speed, it is assumed to be a constant value. However, if there is wind speed data available, varying power rates can also be applied in this approach, as illustrated in [170]. Moreover, there are two predetermined expenses, i.e., the cost to access a turbine C_A and the preparation cost C_{Pre}, associated with the wind turbine and wind farm operations, respectively, which account for the economic dependence of the system. C_A refers to the cost of moving turbine components to a wind turbine when components need to be replaced, and this cost can be intuitively understood as the cost of using a crane [153]. C_{Pre} can be considered as the transportation costs involved in getting a maintenance team to the wind farm and preparing for maintenance.

6.3.5 Formulation of the optimization model

The aim of the CBM optimization is to find optimal maintenance decision variables, i.e., t_m, p_{pm}, and p_{om}, ensuring that the expected annual maintenance cost is minimized while keeping the number of failures within acceptable limits. The maintenance optimization model is formulated by:

$$\begin{aligned} &\min C_E\left(t_m, p_{pm}, p_{om}\right) \\ &s.t.\ \ n_f \le \tau_f \\ &\quad 0 < p_{pm} < p_{om} < 1 \end{aligned} \tag{6.11}$$

where C_E is the expected maintenance cost per unit of time under the proposed CBM policy, n_f is the number of accidental failures of turbine components, and τ_f is the acceptable number of failures. Opportunistic replacement should be executed before the preventive replacement for meaningful solutions. Therefore, two thresholds satisfy $p_{pm} < p_{om}$ and are constrained in the optimization.

6.3.6 Maintenance cost evaluation

It is assumed that the lifetime distributions of all turbine components are known in advance. For the i-th turbine component that can be monitored, it is possible to validate its prognostic error, i.e., $RMSE_i$, by analyzing historical data and to generate a substantial database of prognostic histories, denoted by $D_{e,i}$, which are linked to actual RUL percentages and predicted RUL distributions based on $RMSE_i$, utilizing the method outlined in Section 6.3.2. Specify the maximum simulation time T_{Max}, the simulation time interval Δt, and the maintenance lead time t_L. In addition, for each turbine component, specify the cost values, including the failure maintenance cost $C_{F,i}$, planned inspection cost $C_{I,i}$, and preventive maintenance cost $C_{P,i}$. Two fixed costs, i.e., preparation cost C_{Pre} and the cost to access a turbine C_A, also need to be specified. At the start of the simulation approach, i.e., simulation time $t_S = 0$, the total maintenance cost C_T is set to 0 and will be updated during the simulation process. For turbine component $c_{i,j}$, a real failure time, denoted by $T_{F,i,j}$, is generated from its lifetime distribution. If the component $c_{i,j}$ can be monitored, a prognostic history is randomly sampled from database $D_{e,i}$. Initially, the age values for all the components are 0, that is, $t_{i,j} = 0$ for all i and j.

At a certain time when $t_S > 0$, for the turbine component $c_{i,j}$, its age $t_{i,j}$ and real failure time $T_{F,i,j}$ are known, and the real RUL percentage $r_{i,j}$ is calculated by $(T_{F,i,j} - t_{i,j}) / T_{F,i,j}$. Given the real RUL, the predicted RUL percentage $\hat{r}_{i,j}$ can be inferred from the sampled prognostic history. If the planned inspection is not met, or if the monitorable component $c_{i,j}$ fails to attain the I status during the previous inspection, its posterior RUL percentage is equivalent to the average predicted RUL percentage, i.e., $\hat{r}_{\mathrm{Po},i,j} = \bar{\hat{r}}_{i,j}$. If $c_{i,j}$ is determined to be in either I or II health status, the posterior upward probability is evaluated using Eq. (6.3) to merge inspection approximations by:

$$\mathrm{Po}_{u,i,j} = \frac{P_{i,j}(N_V \mid N_P = u)}{P_{i,j}(N_V \mid N_P = u) + P_{i,j}(N_V \mid N_P = d)} \tag{6.12}$$

where $P_{i,j}(N_V \mid N_P = u)$ and $P_{i,j}(N_V \mid N_P = d)$ are updated conditional probabilities for component $c_{i,j}$ given the current inspection and prognostic information. Based on Eq. (6.4), the predicted failure time is updated to $\hat{t}_{f,\mathrm{Po},i,j}$, thereby updating the current posterior RUL percentage for the turbine component $c_{i,j}$ as:

$$\hat{r}_{\mathrm{Po},i,j} = \frac{\hat{t}_{f,\mathrm{Po},i,j} - t_{i,j}}{\hat{t}_{f,\mathrm{Po},i,j}} \tag{6.13}$$

In the absence of any maintenance action at time t_S, the time point t_S will be shifted to $t_S + \Delta t$, and the component age and actual RUL percentage will be suitably updated. Otherwise, maintenance decisions are made based on the proposed CBM policy presented in Section 6.3.3. In case the turbine component $c_{i,j}$ necessitates replacement, a new real failure time $T_{F,i,j}$ will be generated by drawing a sample from the lifetime distribution of the component subsequent to the completion of maintenance, i.e., after the maintenance lead time plus repair operation time. The component age $t_{i,j}$ and actual RUL percentage $r_{i,j}$ will be reset to zero and one, respectively. In addition, if the component $c_{i,j}$ is monitorable, a prognostic history will be resampled from database $D_{e,i}$ after the component is replaced. Maintenance costs are updated based on three decision variables and the posterior RUL percentage of turbine components under four maintenance scenarios, as illustrated in Eq. (6.14) to Eq. (6.17).

If $t_{i,j} \geq T_{F,i,j}$, the turbine component $c_{i,j}$ is determined to fail, and the component will be replaced after the maintenance lead time. When any component in a wind turbine is replaced, the other

components of that wind turbine will be inspected at the same time. The total cost of failure replacement for the wind farm is given by:

$$\Delta C_{T,F} = \sum_{j=1}^{N}\sum_{i=1}^{M} I_{F,i,j} C_{F,i} + \sum_{j=1}^{N}\sum_{i=1}^{M} I_{F,i,j}^{Insp} C_{I,i}^{S} \tag{6.14}$$

where $I_{F,i,j}$ and $I_{F,i,j}^{Insp}$ are binary variables indicating if the turbine component $c_{i,j}$ fails and if the component $c_{i,j}$ is inspected due to the turbine failure, respectively. If $I_{F,i,j}$ = 1, the failure replacement action will be performed for component $c_{i,j}$, whereas $I_{F,i,j} = 0$ suggests otherwise. $I_{F,i,j}^{Insp}$ = 1 if the turbine component $c_{i,j}$ in the failed turbine j is inspected, and $I_{F,i,j}^{Insp} = 0$ otherwise. $C_{I,i}^{S}$ is the inspection setup cost determined in Eq. (6.10).

The inspection is reached when $t_S = n_I t_m$, where n_I = 1, 2, …, N_I indicates the n_I-th planned inspection, and N_I is the total number of scheduled inspections during the maximum simulation time T_{Max}. If the current time point t_S reaches the inspection time, planned inspection will be performed for all turbines in the wind farm, and the change in the total cost due to inspections is:

$$\Delta C_{T,I} = \sum_{j=1}^{N}\sum_{i=1}^{M} C_{I,i,j} \tag{6.15}$$

where $C_{I,i,j}$ is the planned inspection cost defined in Eq. (6.10). During the planned inspection, if the turbine component $c_{i,j}$ reaches its last in-service health level, i.e., $r_{i,j} < 0.1$, the component will be replaced, and the maintenance cost associated with the component replacement is also considered in the equation.

In case the posterior RUL percentage of a turbine component $c_{i,j}$ falls below the preventive threshold, i.e., $\hat{r}_{\mathrm{Po},i,j} < p_{pm}$, the component is replaced promptly to prevent any potential failure. If a preventive replacement is performed on any of the wind turbine components, a thorough inspection of the remaining components is carried out. The total preventive replacement cost will be incurred for the wind farm as follows:

$$\Delta C_{T,P} = \sum_{j=1}^{N}\sum_{i=1}^{M} I_{P,i,j} C_{P,i} + \sum_{j=1}^{N}\sum_{i=1}^{M} I_{P,i,j}^{Insp} C_{I,i}^{S} \tag{6.16}$$

where $I_{p,i,j} = 1$ if the turbine component $c_{i,j}$ is determined to be preventively replaced, and $I_{p,i,j} = 0$ otherwise. $I_{P,i,j}^{Insp} = 1$ if the component $c_{i,j}$ is inspected due to preventive replacement of any component in the j-th turbine, and $I_{P,i,j}^{Insp} = 0$ otherwise.

The opportunistic replacement decision will be made on the turbine component $c_{i,j}$ when either a) any other component in the j-th wind turbine is determined to be replaced, and $c_{i,j}$ reaches the last in-service health level, i.e., $r_{i,j} < 0.1$, or b) any other wind turbine is being replaced, the component $c_{i,j}$ can be monitored and its posterior RUL percentage falls beneath the opportunistic threshold, i.e., $\hat{r}_{\mathrm{Po},i,j} < p_{om}$. In the second case, not only component $c_{i,j}$ is maintained, but the other components within the j-th wind turbine are also inspected, and any with a RUL of less than 0.1 are replaced as well. If an opportunistic replacement is carried out, the opportunistic replacement costs are incurred. The change in the total cost due to the opportunistic replacement is:

$$\Delta C_{T,O} = \sum_{j=1}^{N}\sum_{i=1}^{M} I_{O,i,j} C_{P,i} + \sum_{j=1}^{N}\sum_{i=1}^{M} I_{O,i,j}^{Insp} C_{I,i,j} \tag{6.17}$$

where $I_{O,i,j}$ and $I_{O,i,j}^{Insp}$ are binary variables. $I_{O,i,j}$ indicates if the turbine component $c_{i,j}$ is opportunistically replaced and $I_{O,i,j}^{Insp}$ denotes whether the component is inspected in the second case of triggering opportunistic replacement. If $I_{O,i,j} = 1$, the opportunistic replacement will be performed for turbine component $c_{i,j}$, whereas $I_{O,i,j} = 0$ suggests otherwise. $I_{O,i,j}^{Insp} = 1$ if the component $c_{i,j}$ is inspected due to opportunistic replacement of components in the j-th turbine, and $I_{O,i,j}^{Insp} = 0$ otherwise.

In the event that a component within a wind farm requires replacement, certain fixed expenses relating to preparation and turbine access will be incurred. Two binary variables I_{Pre} and $I_{A,j}$ are defined to formulate the overall cost. The value of I_{Pre} is set to 1 if maintenance is required for any wind turbine in the wind farm; otherwise, it is set to 0. Similarly, $I_{A,j} = 1$ if replacement actions will be performed on the j-th wind turbine in the wind farm, whereas $I_{A,j} = 0$ suggests otherwise. According to four maintenance cost elements modeled by Eq. (6.14) to Eq. (6.17), the change in the total cost of maintenance at a time point is given by:

$$\Delta C_T = \Delta C_{T,F} + \Delta C_{T,I} + \Delta C_{T,P} + \Delta C_{T,O} + I_{Pre} C_{Pre} + \sum_{j=1}^{N} I_{A,j} C_A \tag{6.18}$$

Consequently, the expected maintenance cost C_E can be derived by referring to the maintenance cost $\Delta C_{T,t_S}$ at each time point as given in Eq. (6.19) upon reaching the maximum simulation time T_{Max}. To account for stochasticity, a multitude of parallel simulations are executed, and the mean value is computed for optimization.

$$C_E = \frac{\sum_{t_S=0}^{T_{Max}} \Delta C_{T,t_S}}{T_{Max}} \tag{6.19}$$

6.4 Case study

6.4.1 System description

This section presents a case study of an offshore wind farm located in the UK that serves as an example to demonstrate the proposed CBM policy. The wind farm comprises 13 identical wind turbines, each of which is a GE Energy 1.5s model rated at 1500 KW. The total power of the wind farm is 19500 KW [143]. The configuration of each wind turbine is the same as the turbines shown in Fig. 5.3 of Chapter 5. The study focuses on analyzing four crucial components in each wind turbine, which include the generator, bearing, gearbox, and rotor-blade. These four components often have a relatively higher likelihood of experiencing failure compared to other components [139]. Hence, for the wind farm under consideration, the value of M is 4, and N is 13.

Table 6.1: Weibull lifetime distributions of major components (time unit: month)

Component	Scale parameter α	Shape parameter β
Generator	190	2
Bearing	275	3
Gearbox	140	2.5
Rotor-blade	205	3.5

In the wind turbines under consideration, the main bearing and gearbox are subject to online monitoring via accelerometers, as their degradation is usually accompanied by an increase in the vibration amplitude. Consequently, real-time prognostic models developed from monitored vibrations can predict the RUL of both components. In contrast, the rotor-blade and generator in the studied wind turbine lack sensors to measure their health conditions. Manual techniques, such as non-destructive testing, are necessary to detect the health of these components during planned

inspection or maintenance. The Weibull distribution is employed to describe the failure times of turbine components. Table 6.1 summarizes the Weibull shape and scale parameters for each critical component [10]. It is noted that the choice of distribution assumption can introduce errors and cause the model to deviate from reality. The failure time distribution of turbine components is unique to the particular working conditions. Therefore, it is crucial to estimate the distribution as much as possible utilizing failure or right-censored data of turbines specific to the target operating environment.

Table 6.2: Maintenance cost data (cost unit: $k)

Component	Part cost	Failure setup cost	Inspection setup cost	Preventive setup cost	Cost to access a turbine	Preparation cost
Generator	92	166.5	3.735	37.35		
Bearing	31	166.5	3.735	37.35		
Gearbox	155	181.44	4.482	44.82	44	50
Rotor-blade	88	151.56	2.988	29.88		

The cost data of wind turbine components are primarily based on NREL reports [171][172], supplemented by additional works in [153][173][174]. Table 6.2 presents the maintenance costs for every part of the turbine. The labor cost rate for maintenance and inspection is fixed at $80 per hour, with two laborers earning $40 per hour each [153]. The electricity price is determined to be $85/MWH, leading to a power loss rate of $127.5 per hour for a turbine power output of 1.5 MW [173]. The cost to access a turbine C_A is regarded to be the expense of utilizing a crane, which is calculated according to [171]. The fixed preparation cost C_{Pre} is specified as a considerably high value, as reported in [153], in order to account for the economic dependence of wind farms when dealing with challenging operating conditions.

The downtime required for preventive replacement of the generator, bearing, gearbox, and rotor-blade is estimated to be 0.25, 0.25, 0.3, and 0.2 months, respectively [153][174]. The duration for failure replacement and inspection is assumed to be 2 times and 0.1 times the preventive replacement time, respectively. These rough estimates of replacement time may not perfectly match real-world situations. However, it is possible to build a more realistic simulation environment by assigning different probabilities and repair times to turbine components

corresponding to their failure modes. The setup cost includes labor costs and revenue loss incurred due to downtime. The maintenance lead time is assumed to be 1 month. The setup cost for failure replacement also includes the revenue loss caused by the maintenance lead time, as the maintenance actions are performed after the preparation.

6.4.2 Comparative policies and simulation settings

In this work, five additional policies are formulated for the maintenance decision-making of the wind farm. The first two policies are CM and a TBM policy, i.e., constant interval maintenance (CI), respectively, which are widely used in wind farm maintenance management when condition monitoring data is not available. The action of CM is performed only when a component fails. Therefore, the maintenance cost model of CM can be expressed as Eq. (6.20). CI policies determine maintenance actions based on a constant planned inspection interval, see [151] for details. In CI policies, an extra cost item is included to account for the planned inspection. The maintenance cost model for CI can be formulated by Eq. (6.21).

$$\min C_E^{CM} = \frac{\sum_{t_S=0}^{T_{Max}} \left(\sum_{j=1}^{N} \sum_{i=1}^{M} I_{F,i,j} C_{F,i} + I_{Pre} C_{Pre} + \sum_{j=1}^{N} I_{A,j} C_A \right)}{T_{Max}} \tag{6.20}$$

$$\min C_E^{CI} = \frac{\sum_{t_S=0}^{T_{Max}} \left(\sum_{j=1}^{N} \sum_{i=1}^{M} I_{F,i,j} C_{F,i} + \Delta C_{T,I} + I_{Pre} C_{Pre} + \sum_{j=1}^{N} I_{A,j} C_A \right)}{T_{Max}} \tag{6.21}$$

The comparative methods referred to as CBM1, CBM2, and CBM3, respectively, are the last three CBM methods evaluated in this study. Three CBM policies consider opportunistic maintenance the same as this work. In CBM1 and CBM2, maintenance actions will be executed if the average predicted RUL percentage is less than the maintenance thresholds. In contrast, CBM3 finds maintenance thresholds based on the failure probability given by Eq. (6.22) [161]. In this formulation, $f_{fail}(x)$ is the probability distribution function of the predicted failure time, and t_L is the maintenance lead time. CBM3 performs preventive or opportunistic replacement for turbine components if the predicted failure probability is higher than the corresponding threshold. CBM1 modified from [74] uses two maintenance thresholds without considering the planned inspection, while CBM2 and CBM3 are developed from policies proposed in Chapter 5 to consider economic

dependency among wind turbines in combination with planned inspections. The maintenance cost model for CBM1 is formulated by Eq. (6.23), and CBM2 and CBM3 are modeled by Eq. (6.24).

$$\mathrm{P}_f = \frac{\int_t^{t+t_L} f_{fail}(x)dx}{\int_t^{\infty} f_{fail}(x)dx} \tag{6.22}$$

$$\min C_E^{CBM1} = \frac{\sum_{t_S=0}^{T_{Max}} \left(\Delta C_{T,F} + \Delta C_{T,P} + \Delta C_{T,O} + I_{Pre} C_{Pre} + \sum_{j=1}^{N} I_{A,j} C_A \right)}{T_{Max}} \tag{6.23}$$

$$\min C_E^{CBM2,CBM3} = \frac{\sum_{t_S=0}^{T_{Max}} \left(\Delta C_{T,F} + \Delta C_{T,I} + \Delta C_{T,P} + \Delta C_{T,O} + I_{Pre} C_{Pre} + \sum_{j=1}^{N} I_{A,j} C_A \right)}{T_{Max}} \tag{6.24}$$

In the simulation, the maximum simulation time T_{Max} is assumed to be 1000 months, and the simulation time interval Δt is set to 1 month. We limit the number of turbine component failures to 0.3 per year. Given a set of maintenance decision variables, 1×10^3 simulations are performed in parallel, and the average value of maintenance cost C_E is calculated. The simulation time and parallelism number are large enough to obtain stable solutions while considering distribution randomness.

6.4.3 Maintenance optimization with informative prognostics

We start discussing the benefits of the proposed maintenance policy with accurate prognostic information. In this scenario, we assume that the RMSE falls between 0.03 and 0.06, which is able to provide reliable prognostic information in maintenance scheduling. By utilizing the proposed policy, the optimal combination of the maintenance decision variables is found to be t_m = 19 months, p_{pm} = 0.4, and p_{om} = 0.9, and the corresponding expected maintenance cost C_E is \$78.1887k/month. Fig. 6.4 shows the plot of maintenance cost versus t_m while keeping p_{pm} at 0.4 and p_{om} at 0.9. The expected maintenance cost as a function of the preventive threshold p_{pm} and the opportunistic threshold p_{om} with t_m fixed at 19 months is presented in Fig. 6.5. When the number of turbine component failures exceeds the permitted limit, a penalty is added to the total maintenance cost. As can be seen in Fig. 6.4 and Fig. 6.5, the maintenance cost model has clear concavity both in terms of profile and surface. This means that there is always a combination of

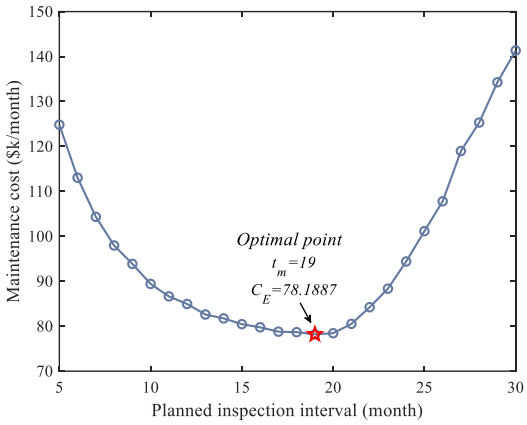


Fig. 6.4: Maintenance cost versus planned inspection interval t_m

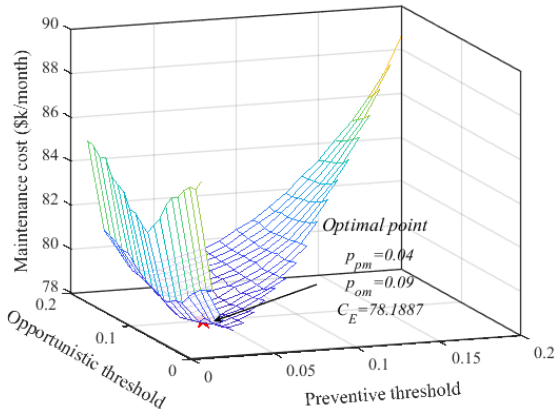


Fig. 6.5: Maintenance cost versus different combinations of predictive maintenance thresholds

maintenance decisions that can minimize the maintenance cost of a wind farm. In addition, the result suggests that it is cheaper to schedule maintenance taking into account planned inspections, rather than performing actions based only on two maintenance thresholds, due to the monitorability and prognostic error of different turbine components and the economic dependence among wind turbines.

Table 6.3 lists the performance of the proposed policy and the five comparative maintenance policies in terms of reducing maintenance costs using informative prognostics. The TBM policy is straightforward to implement since only t_m needs to be decided but may result in periodic unnecessary maintenance actions. Prognostics with high accuracy provides reliable information about component failure, allowing for maintenance actions to be taken just before failure occurs and achieving zero-downtime performance. Meanwhile, informative prognostics supports

preventive maintenance on other turbine components when a maintenance opportunity arises, leading to more effective scheduling of maintenance actions. The proposed policy shows significant cost reductions compared to other policies, ranging from 0.16% to 21.39%. Since the wind turbine is partially monitorable, maintenance decisions based solely on prognostics, i.e., the CBM1 policy, require a higher opportunistic threshold to meet failure times limits. Considering planned inspections, as seen in the cost comparison between CBM1 and CBM2, can yield better performance. Overall, informative prognostics enables different maintenance bases, including average RUL percentage in CBM2, failure probability in CBM3, and posterior RUL percentage in the proposed method, to achieve similar benefits for the wind farm.

Table 6.3: Comparison of different maintenance policies with informative prognostics (time unit: month, cost unit: $k)

Policies	Maintenance decision variables	Maintenance cost	Reduction
CM	–	99.4699	21.39%
CI	$t_m = 13$	84.2820	7.23%
CBM1	$p_{pm} = 0.02$, $p_{om} = 0.77$	82.4339	5.15%
CBM2	$t_m = 21$, $p_{pm} = 0.02$, $p_{om} = 0.06$	78.5366	0.44%
CBM3	$t_m = 21$, $p_{pm} = 0.81$, $p_{om} = 0.09$	78.3123	0.16%
Proposed policy	$t_m = 19$, $p_{pm} = 0.4$, $p_{om} = 0.9$	78.1887	–

6.4.4 Maintenance optimization considering prognostic errors

Fig. 6.6 displays the optimized maintenance cost of the proposed policy and three comparative CBM policies, including CBM1, CBM2, and CBM3, with different prognostic errors. When prognostic error is relatively small (RMSE < 0.06), the difference in benefit between different CBM policies is not significant. However, it is noted that in real-world applications, the RMSE of prognostic models typically falls within the range of 0.1 to 0.15 [126]. In data-scarce situations, the prognostic error may be even higher, as shown in the results of Chapter 2. As the prognostic error is increased to the range of 0.09 to 0.12, the proposed policy demonstrates remarkable superiority, resulting in maintenance cost reductions of around 11.63%, 4.40%, and 2.09%, respectively, versus CBM1, CBM2, and CBM3 policies. More convincingly, when there are large biases in RUL predictions with RMSE ranging from 0.15 to 0.18, the new maintenance basis

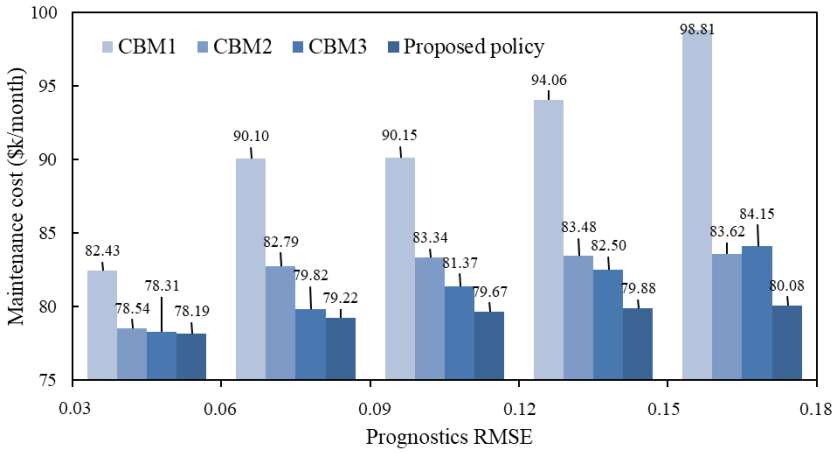


Fig. 6.6: Comparison of CBM policies with different prognostic errors

allows for more efficient decision-making, leading to cost reductions of up to 18.96%, 4.23%, and 4.84% compared to other CBM policies.

In these comparative policies, the maintenance cost increases rapidly with prognostic errors under the CBM1 policy. This is because CBM1 relies solely on prognostic information. When there are significant errors in prognostics, CBM1 sets higher predictive thresholds to prevent potential failures. As a result, premature replacement of turbine components is triggered even when they are still in good condition. In contrast, CBM2 and CBM3 policies optimize both predictive thresholds and planned inspection intervals. When there are large errors, the predictive thresholds of CBM2 and CBM3 policies tend to be zero, and they no longer determine maintenance actions. However, maintenance decisions are still influenced by inspections under CBM2 and CBM3 policies, resulting in relatively stable maintenance costs that stay below $84.2820k/month, as listed in CI cost of Table 6.3.

The proposed maintenance basis utilizes turbine health estimates gathered during inspections to calibrate RUL predictions in the presence of inaccurate prognostics. It is noted that even in the worst-case scenario where the model parameters in Eq. (6.2) are not properly configured, the optimized decisions and benefits of the proposed policy are at least comparable to those of CBM2. This is because the proposed policy incorporates time-based maintenance decisions, which allows for fundamental benefits of maintenance optimization and a degree of fault tolerance in relation to parameter settings. In previous studies [92][161], prognostic results were represented as normal distributions with zero mean and constant variance. This is a robust optimization strategy that can

mitigate the potential impacts of prognostic uncertainties. However, such a strategy is unable to account for prediction bias, so the optimized decisions are not robust when facing actual error situations. In cases of significant prognostic errors, the maintenance approaches used in previous studies [92][161] yield almost similar benefits as CBM2 and CBM3 policies.

6.4.5 Sensitivity analysis

Various practical factors have been considered in previous studies with respect to their impacts on the benefits of wind farm maintenance. Many of these factors have a direct influence on maintenance decisions as they change the values of related parameters in the optimization model, e.g., maintenance lead time [92], fixed preparation cost [153], and part costs [10]. The maintenance decisions also depend on the age of the turbines and the manufacturer, which contribute to the failure time distribution of turbine components. Some factors can be considered by including additional indicators in the maintenance optimization. For instance, a reduction indicator is defined when resampling the lifetime of components after they have been replaced in order to reflect imperfect maintenance in Section 5.4.3. There are also factors that not only affect maintenance optimization, but also RUL predictions, such as weather conditions and operational conditions. These two factors introduce different environmental disturbances and non-linear conditions to prognostics, and they also impact the preparation costs and the failure time distributions for turbine components. Additionally, wind speed influences the power generation rate of wind turbines and is linked to the revenue loss caused by downtime [170]. As a result, maintenance decisions should be made on a case-by-case basis.

In this section, the impact of the proposed maintenance basis driven by different health estimation biases from inspection on CBM benefits is further investigated. To this aim, a parameter δ is defined as a measure of inspection biases. When the RUL percentage of a turbine component deviates from the predefined boundary conditions of I status or II status by δ, the component still has a half probability of being inspected in the corresponding health status. Fig. 6.7 presents the maintenance cost of the proposed policy versus different inspection biases δ with the RMSE of prognostics from 0.15 to 0.18. When the inspection bias is less than 0.02, the maintenance cost curve is flatter relative to the δ value. In this region, even a 50% decline in the performance of inspection estimates, i.e., δ from 0.02 to 0.01, will result in only a 0.26% reduction in the cost. In contrast, when the inspection bias is from 0.02 to 0.08, the maintenance cost shows a rapid rise

with increasing δ. After the inspection bias δ is greater than 0.08, the trend of maintenance cost increases with δ tends to flatten again. Although the new maintenance basis can be affected by inspection estimates, the proposed policy still results in about 16.66%, 1.53%, and 2.15% reduction in maintenance costs compared to CBM1, CBM2, and CBM3, respectively, even with a large δ of 0.1. Therefore, the proposed maintenance basis connects prognostics and inspection information, thus compensating each other for the impact of their uncertainties on maintenance benefits to find better maintenance plans.

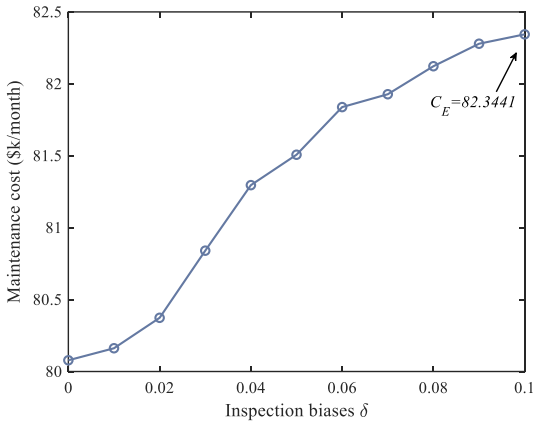


Fig. 6.7: Maintenance cost versus health estimation biases δ

6.5 Conclusions

This chapter focuses on optimizing maintenance decisions for wind farms with imperfect prognostic information. We consider two types of turbine components and assume that only specific parts of the wind turbine are monitored. In addition to predictive thresholds, the planned inspection time is also considered as an optimization candidate. A new maintenance basis, defined by the posterior RUL percentage, is proposed to determine maintenance actions in combination with prognostic results and inspection information. The proposed policy calibrates RUL predictions based on inspection information, leading to enhanced maintenance benefits. As discussed in the case study, the proposed method can reduce maintenance costs by at least 4% when there are significant errors in prognostics. Sensitivity analysis demonstrates that even if the inspection information is biased, the proposed maintenance strategy still outperforms other policies. Consequently, this study has practical implications for reducing the operation and maintenance expenses of wind farms.

Due to the practical nature of wind turbines, they may encounter various failure modes and unexpected situations. These real-world factors cannot be fully captured in the maintenance optimization model. Therefore, it is essential to introduce randomness into the system maintenance models to accommodate unknown uncertainties. In this thesis, maintenance scheduling uncertainty is addressed in two ways. First, it is assumed that the lifetime distributions of all components are known in advance. During simulation, when a component in a wind turbine is replaced, a new lifetime is generated by sampling from its lifetime distribution, thereby introducing uncertainty caused by machine failure into the maintenance optimization. Second, the RUL curves are simulated based on the exponential model with stochastic parameters. For components that can be monitored, a prognostic history is randomly generated by the proposed model, introducing uncertainty due to prognostics in maintenance decision-making. Furthermore, the proposed maintenance basis has a wide range of real-world applications besides wind farms, as in most systems, the prognostic information is incomplete and not sufficiently accurate. However, the maintenance policies in this study are designed specifically for wind farms. For other systems, remodeled maintenance costs are necessary. Future research can be directed towards developing a simplified strategy to ensure robust maintenance decisions in the face of prognostic errors without requiring RUL calibration and quantifying the inter-dependency of RUL between components in a wind turbine into the maintenance optimization model.

Chapter 7: Summary and future work

In this chapter, the novel contributions of the research work conducted in this thesis are summarized in Section 7.1. In addition, several research problems that are worth to be explored in the future are presented in Section 7.2.

7.1 Summary

To effectively plan maintenance schedules using prognostic information, it is crucial to address real-world issues at both the component and system levels. Artificial intelligence techniques, particularly deep learning methods, serve as a connection between monitoring data and intelligent asset maintenance. These state-of-art techniques are gaining increasing attention in prognostics and health management due to their exceptional data processing capabilities, and they have the potential to address the aforementioned concerns for improved RUL predictions and maintenance activities. The main objective of this thesis is to develop deep learning models for prognostics under various scenarios of data availability, and the maintenance decisions are then made at the system level by incorporating prognostic information. This thesis makes two main contributions in terms of component-level prognostics and system-level maintenance, which are summarized as follows.

7.1.1 Prognostics with limited available historical data

If sufficient condition data is collected, various data-driven models can be employed to make accurate RUL predictions. However, in reality, engineering assets are often replaced before they fail, which means only a limited number of failure records can be gathered for modeling purposes. In the first three topics, deep learning techniques are developed to tackle the challenge of limited data availability in prognostics.

Specifically, in Chapter 2 of this thesis, we focus on situations where only a few instances of failure data but a larger amount of suspension data can be collected for model training. In this study, a semi-supervised learning method is proposed that uses both failure and suspension histories to predict the RUL of machines. Unlike previous approaches, the loss function of our model is

established using a multi-task objective function that captures valuable information from suspension histories, instead of treating them as unlabeled data. This allows for making more accurate estimates of RUL that align with physical expectations based on both failure and suspension data.

Chapter 3 of the thesis focuses on the prognostics of machines operating under different conditions. The target machine in this study has new properties and operating conditions compared to the machine recorded in the historical data. This leads to significant differences in their measurement distributions. We consider that there is a small amount of suspension histories available to be collected during planned inspections when the machine operates in a new environment. A novel method that combines COSMO and transfer learning is proposed to utilize the limited suspension histories from the target domain to enhance the generalization of the model to new environments. To this aim, a joint MMD method incorporating manifold regularization is proposed. This method builds upon previous studies by adapting monitoring features from different domains to ensure consistent degradation trajectories from both marginal and conditional perspectives. Consequently, it significantly improves the predictions of RUL using the limited suspension histories available.

In Chapter 4 of this thesis, the focus is on predicting the RUL of machines that are working under new conditions. However, it is not feasible to gather any form of data from machines operating under these conditions. To build models under this scenario, a new RUL prediction method based on state-space modeling and reinforcement learning is presented. This model is developed with Lyapunov constraints within the framework of reinforcement learning and is trained using the adversarial strategy based on the concept of $H\infty$ robustness. This allows for improving the robustness of the model to different conditions by reducing the causality between RUL and operating parameters and increasing the causality between RUL and unobserved degradation characteristics. Therefore, the proposed model is both interpretable and capable of predicting RUL for equipment operating beyond historical records.

7.1.2 Maintenance decision-making using partial and inaccurate prognostics

Recent developments in prognostic models have led to increased use of prognostic information to improve the benefits of CBM. The last two topics have explored the more practical integration of prognostics into decision-making for system maintenance. Chapter 5 of this thesis aims to address

the limitations of previous studies, which assumed that prognostic information is available for all components and neglected the economic loss caused by system degradation when assessing the benefits. This is achieved by proposing a prognostics-induced CBM policy that finds a joint maintenance decision with maximum net revenue. This decision involves optimizing maintenance intervals and predictive thresholds, considering that prognostics is only available for certain components. In addition, a data-driven efficiency modeling method is incorporated into the net revenue optimization framework to account for the economic loss resulting from system degradation when assessing the maintenance benefits. The maintenance costs and policies are formulated into a large-scale optimization model to capture the economic dependencies within the system. Insights on the influence of prognostic error on maintenance costs have also been derived by error modeling and sensitivity analysis.

Chapter 6 of the thesis upon the work in Chapter 5 by developing maintenance optimization strategies for a more realistic system, specifically a wind farm. The contribution of this study is formed on addressing the challenges of maintenance scheduling for wind farms. These challenges are highlighted not only by high setup costs and strong economic dependencies, but more significantly by partial monitorability and potentially inaccurate prognostic information. To this aim, a new maintenance basis denoted by the posterior RUL percentage is presented to determine preventive maintenance actions. This maintenance basis integrates auxiliary health estimates gathered during inspections with predictive analytics to calibrate RUL predictions, allowing for better decision-making to enhance maintenance benefits, especially in situations where prognostics is imperfect.

7.2 Future work

Although the studies conducted in this thesis have overcome the limitations of previous research on prognostics and system maintenance optimization, there still exist some challenges and problems that need further exploration, which are described as follows.

7.2.1 Incorporation of physical mechanisms for prognostics and health management

In this thesis, some data-driven methods have been proposed for prognostics. These methods rely on degradation models or past failure histories, but they do not fully consider the dynamic nature

of the target machines. As a result, they have limitations in explaining the degradation mechanisms and providing accurate predictions when no failure data is available. In other words, if there is no failure history for model training, the methods discussed in Chapters 2, 3, and 4 cannot be used. Typical physical information includes quantities that reflect machine degradation, such as crack length, wear mass, structural stiffness loss, etc. In order to acquire this information, these physical quantities must be simulated to anticipate their patterns over time. For instance, the crack length can be predicted utilizing a degradation model known as "Paris' law". Other examples include forecasting wear volume with "Archard's model" and predicting time-varying stiffness loss in gearbox dynamics with various failure modes. The integration of physical mechanisms to guide the model training of data-driven techniques deserves further investigation.

7.2.2 Prognostics assessment from the perspective of maintenance benefits

Prognostics is commonly employed to assist in developing maintenance plans for mechanical systems. However, most prognostic methods focus on the accuracy of their predictions for individual components, neglecting to analyze the cost-benefit of using these predictions in system maintenance. Many companies are interested in understanding the potential benefits of using prognostic models for preventive maintenance. Instead of just providing a RMSE value like in Chapter 2, 3, and 4, it is more acceptable to present the benefits of a prognostic model as a distribution that demonstrates the reduction in maintenance cost along with the corresponding probability. While it is ideal to utilize prognostics with minimal errors for maximum maintenance benefits, blindly optimizing prognostic models in situations with limited historical data can be costly and ineffective. Even with the best prognostic model in such cases, the improvement in maintenance benefits may still be insignificant. A possible solution is to develop methods that link specific prognostic models to the maintenance optimization process and evaluate their cost-effectiveness. To this aim, the key idea is to create a generative model that simulates predicted RUL profiles with features that closely resemble the specified prognostic model on the test dataset. This approach will be further explored in future works.

7.2.3 Maintenance decision-making with both prognostics and diagnostics

Chapters 5 and 6 of this thesis have investigated the potential use of prognostics to guide maintenance plans for a large mechanical system, even when the prognostic information is

incomplete and not sufficiently accurate. However, the possible different failure modes of the components are not considered. The impact of failure modes on system maintenance can be seen in two ways: the cost and necessity of maintenance actions. If a failure could have serious consequences, maintenance should be carried out immediately. On the other hand, if the impact of a failure is minor, it is often possible to wait until the next scheduled maintenance before overhauling the various components of the system together. To achieve this, diagnostics should also be conducted to assist in making maintenance decisions. In most cases, it is not feasible to obtain a completely clear diagnosis of the failure mode. Instead, risk indicators for failure modes can be analyzed. A multi-objective maintenance decision-making approach can be developed to strike a balance between avoiding risks and minimizing maintenance costs.

References

[1] ISO 13372: 2012. Condition monitoring and diagnostics of machines – Vocabulary. International Organization for Standardization, ISO.

[2] Cempel C. Multidimensional condition monitoring of mechanical systems in operation. Mechanical Systems and Signal Processing, 2003; 17 (6): 1291–1303.

[3] Salgado J P M, Sanz J M, Ramírez B A, Cordero C A. Identification of the mechanical failure factors with potential influencing road accidents in Ecuador. International Journal of Environmental Research and Public Health, 2022; 19 (13): 7787.

[4] Hamilton S D, Millstein D, Bolinger M, Wiser R, Jeong S. How does wind project performance change with age in the United States? Joule, 2020; 4 (5): 1004–1020.

[5] Staffell I, Green R. How does wind farm performance decline with age? Renewable Energy, 2014; 66: 775–786.

[6] Zio E, Compare M. Evaluating maintenance policies by quantitative modeling and analysis. Reliability Engineering & System Safety, 2013; 109: 53–65.

[7] Waeyenberg G, Pintelon L. A framework for maintenance concept development. International Journal of Production Economics, 2002; 77 (3): 299–313.

[8] Bevilacqua M, Braglia M. The analytic hierarchy process applied to maintenance strategy selection. Reliability Engineering & System Safety, 2000; 70 (1): 71–83.

[9] Cherkaoui H, Huynh K T, Grall A. Quantitative assessments of performance and robustness of maintenance policies for stochastically deteriorating production systems. International Journal of Production Research, 2018; 56 (3): 1089–1108.

[10] Yang L, Li G, Zhang Z, Ma X, Zhao Y. Operations & maintenance optimization of wind turbines integrating wind and aging information. IEEE Transactions on Sustainable Energy, 2021; 12 (1): 211–221.

[11] Heng A, Zhang S, Tan A C C, Mathew J. Rotating machinery prognostics: State of the art, challenges and opportunities. Mechanical Systems and Signal Processing, 2009; 23 (3): 724–739.

[12] Schwabacher M A. A Survey of Data-Driven Prognostics. Proceedings of AIAA Infotech @ Aerospace Conference, 2005, 1–5.

[13] Yumbe Y, Hasegawa T, Furukawa N. Inspection schedule optimization technique for power distribution facilities. IEEE Transactions on Power Delivery, 2013; 28: 1558–1565.

[14] Jonge B, Teunter R, Tinga T. The influence of practical factors on the benefits of condition-based maintenance over time-based maintenance. Reliability Engineering & System Safety, 2017; 158: 21–30.

[15] Goyal D, Vanraj, Pabla B S, Dhami S S. Condition monitoring parameters for fault diagnosis of fixed axis gearbox: A review. Archives of Computational Methods in Engineering, 2017; 24: 543–556.

[16] Lei Y, Li N, Guo L, Li N, Yan T, Lin J. Machinery health prognostics: A systematic review from data acquisition to RUL prediction. Mechanical Systems and Signal Processing, 2018; 104: 799–834.

[17] Jardine A K S, Lin D, Banjevic D. A review on machinery diagnostics and prognostics implementing condition-based maintenance. Mechanical Systems and Signal Processing, 2006; 20: 1483–1510.

[18] Baker Hughes develops predictive maintenance software for gas and oil extraction equipment using data analytics and machine learning. https://www.mathworks.com/ company/user_ stories/baker-hughes-develops-predictive-maintenance-software-for-gas-and-oil-extraction-equipment-using-data-analytics-and-machine-learning.html?s_tid= srchtitle_positive%20dis placement%20pumps%20truck_1.

[19] Dupuis R. Oil debris monitoring for managing wind turbine gearbox maintenance. CanWEA's 24. annual conference and trade show: fast forward to wind, Canada, 2008, 19–22.

[20] Niu G, Yang B S, Pecht M. Development of an optimized condition-based maintenance system by data fusion and reliability-centered maintenance. Reliability Engineering & System Safety, 2010; 95: 786–796.

[21] Khan S, Yairi T. A review on the application of deep learning in system health management. Mechanical Systems and Signal Processing, 2018; 107: 241–265.

[22] Tian Z, Wong L, Safaei N. A neural network approach for remaining useful life prediction utilizing both failure and suspension histories. Mechanical Systems and Signal Processing, 2010; 24: 1542–1555.

[23] Jardine A K S, Tsang A H C. Maintenance, replacement, and reliability: Theory and applications. CRC/Taylor & Francis, Boca Raton, 2006.

[24] Lu C, Meeker W. Using degradation measures to estimate a time-to-failure distribution. Technometrics, 1993; 35 (2): 161–174.

[25] Wilson A G, Anderson-Cook C M, Huzurbazar A V. A case study for quantifying system reliability and uncertainty. Reliability Engineering & System Safety, 2011; 96 (9): 1076–1084.

[26] Soliman A A, Abd-Ellah A H, Abou-Elheggag N A, Ahmed E A. Modified Weibull model: A Bayes study using MCMC approach based on progressive censoring data. Reliability Engineering & System Safety, 2012; 100: 48–57.

[27] Kim S, Kim N H, Choi J H. Prediction of remaining useful life by data augmentation technique based on dynamic time warping. Mechanical Systems and Signal Processing, 2020; 136: 106486.

[28] Porotsky S, Bluvband Z. Prognostic and health management for suspended time-series, 2016 second international symposium on stochastic models in reliability engineering, Life Science and Operations Management (SMRLO), Beer Sheva, Israel, 2016, 81–86.

[29] Lu C, Tao L, Fan H. An intelligent approach to machine component health prognostics by utilizing only truncated histories. Mechanical Systems and Signal Processing, 2014; 42 (1–2): 300–313.

[30] Xiao L, Chen X, Zhang X, Liu M. A novel approach for bearing remaining useful life estimation under neither failure nor suspension histories condition. Journal of Intelligent Manufacturing, 2017; 28: 1893–1914.

[31] Zhao R, Yan R, Chen Z, Mao K, Wang P, Gao R X. Deep learning and its applications to machine health monitoring. Mechanical Systems and Signal Processing, 2019; 115: 213–237.

[32] Fink O, Wang Q, Svensén M, Dersin P, Lee W J, Ducoffe M. Potential, challenges and future directions for deep learning in prognostics and health management applications. Engineering Applications of Artificial Intelligence, 2020; 92: 103678.

[33] Zhu X. Semi-supervised learning literature survey. Computer Science, 2008; 37 (1): 63–77.

[34] Hu C, Youn B D, Kim T, Wang P. A co-training-based approach for prediction of remaining useful life utilizing both failure and suspension data. Mechanical Systems and Signal Processing, 2015; 62–63: 75–90.

[35] Ellefsen A L, Bjørlykhaug E, Æsøy V, Ushakov S, Zhang H. Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. Reliability Engineering and System Safety, 2019; 183: 240–251.

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

[37] You M Y, Meng G. A framework of similarity-based residual life prediction approaches using degradation histories with failure, preventive maintenance, and suspension events. IEEE Transactions on reliability, 2013; 62 (1): 127–135.

[38] Kostopoulos G, Karlos S, Kotsiantis S, Ragos O. Semi-supervised regression: A recent review. Journal of Intelligent & Fuzzy Systems, 2018; 35 (2): 1483–1500.

[39] Zhao F, Tian Z, Bechhoefer E, Zeng Y. An integrated prognostics method under time-varying operating conditions. IEEE Transactions on Reliability, 2015; 64 (2): 673–686.

[40] Rigamonti M, Baraldi P, Zio E, Astigarraga D, Galarza A. Particle filter-based prognostics for an electrolytic capacitor working in variable operating conditions. IEEE Transactions on Power Electronics, 2016; 31 (2): 1567–1575.

[41] Wang J, Zhang L, Zheng Y, Wang K. Adaptive prognosis of centrifugal pump under variable operating conditions. Mechanical Systems and Signal Processing, 2019; 131: 576–591.

[42] Zhu L, Chen J. Prognostics of PEM fuel cells based on Gaussian process state space models. Energy, 2018; 149: 63–73.

[43] Wang Y, Peng Y, Zi Y, Jin X, Tsui K. A two-stage data-driven-based prognostic approach for bearing degradation problem. IEEE Transactions on Industrial Informatics, 2016; 12 (3): 924–932.

[44] Wang Z Q, Hu C H, Si X S, Zio E. Remaining useful life prediction of degrading systems subjected to imperfect maintenance: Application to draught fans. Mechanical Systems and Signal Processing, 2018; 100: 802–813.

[45] Liu J, Wang W, Ma F, Yang Y B, Yang C S. A data-model-fusion prognostic framework for dynamic system state forecasting. Engineering Applications of Artificial Intelligence, 2012; 25 (4): 814–823.

[46] Baraldi P, Compare M, Sauco S, Zio E. Ensemble neural network-based particle filtering for prognostics. Mechanical Systems and Signal Processing, 2013; 41 (1–2): 288–300.

[47] Kundu P, Darpe A K, Kulkarni M S. Weibull accelerated failure time regression model for remaining useful life prediction of bearing working under multiple operating conditions. Mechanical Systems and Signal Processing, 2019; 134: 106302.

[48] Li Z, Zheng Z, Outbib R. Adaptive prognostic of fuel cells by implementing ensemble echo state networks in time-varying model space. IEEE Transactions on Industrial Electronics, 2020; 67 (1): 379–389.

[49] Li N, Gebraeel N, Lei Y, Bian L, Si X. Remaining useful life prediction of machinery under time-varying operating conditions based on a two-factor state-space model. Reliability Engineering & System Safety, 2019; 186: 88–100.

[50] Wang H, Liao H, Ma X. Remaining useful life prediction considering joint dependency of degradation rate and variation on time-varying operating conditions. IEEE Transactions on Reliability, 2021; 70 (2): 761–774.

[51] Ren L, Sun Y, Cui J, Zhang L. Bearing remaining useful life prediction based on deep Autoencoder and deep neural networks. Journal of Manufacturing Systems, 2018; 48: 71–77.

[52] Guo L, Li N, Jia F, Lei Y, Lin J. A recurrent neural network based health indicator for remaining useful life prediction of bearings. Neurocomputing, 2017; 240: 98–109.

[53] Li X, Ding Q, Sun J Q. Remaining useful life estimation in prognostics using deep convolution neural networks. Reliability Engineering and System Safety, 2018; 172: 1–11.

[54] Zhai S, Gehring B, Reinhart G. Enabling predictive maintenance integrated production scheduling by operation-specific health prognostics with generative deep learning. Journal of Manufacturing Systems, 2021; 61: 830–855.

[55] Huang C G, Huang H Z, Li Y F. A bidirectional LSTM prognostics method under multiple operational conditions. IEEE Transactions on Industrial Electronics, 2019; 66 (11): 8792–8802.

[56] Liu C, Zhang L, Yao R, Wu C. Dual attention-based temporal convolutional network for fault prognosis under time-varying operating conditions. IEEE Transactions on Instrumentation and Measurement, 2021; 70: 3512210.

[57] Fan Y, Nowaczyk S, Rögnvaldsson T. Transfer learning for remaining useful life prediction based on consensus self-organizing models. Reliability Engineering and System Safety, 2020; 203: 107098.

[58] Sun C, Ma M, Zhao Z, Tian S, Yan R, Chen X. Deep transfer learning based on sparse Autoencoder for remaining useful life prediction of tool in manufacturing. IEEE Transactions on Industrial Informatics, 2019; 15 (4): 2416–2425.

[59] da Costa P R O, Akçay A, Y. Zhang, U. Kaymak, Remaining useful lifetime prediction via deep domain adaptation. Reliability Engineering and System Safety, 2020; 195: 106682.

[60] Zeng F, Li Y, Jiang Y, Song G. An online transfer learning-based remaining useful life prediction method of ball bearings. Measurement, 2021; 176: 109201.

[61] Mao W, He J, Zuo M J. Predicting remaining useful life of rolling bearings based on deep feature representation and transfer learning. IEEE Transactions on Instrumentation and Measurement, 2020; 69 (4): 1594–1608.

[62] Ding P, Jia M, Zhao X. Meta deep learning based rotating machinery health prognostics toward few-shot prognostics. Applied Soft Computing Journal, 2021; 104: 107211.

[63] Zhu J, Chen N, Shen C. A new data-driven transferable remaining useful life prediction approach for bearing under different working conditions. Mechanical Systems and Signal Processing, 2020; 139: 106602.

[64] Cheng H, Kong X, Chen G, Wang Q, Wang R. Transferable convolutional neural network based remaining useful life prediction of bearing under multiple failure behaviors. Measurement, 2021; 168: 108286.

[65] Ding Y, Jia M, Miao Q, Huang P. Remaining useful life estimation using deep metric transfer learning for kernel regression. Reliability Engineering and System Safety, 2021; 212: 107583.

[66] Fu S, Zhang Y, Lin L, Zhao M, Zhong S. Deep residual LSTM with domain-invariance for remaining useful life prediction across domains. Reliability Engineering and System Safety, 2021; 216: 108012.

[67] Li X, Zhang W, Ma H, Luo Z, Li X. Data alignments in machinery remaining useful life prediction using deep adversarial neural networks. Knowledge-Based Systems, 2020; 197: 105843.

[68] Zhang W, Li X, Ma H, Luo Z, Li X. Transfer learning using deep representation regularization in remaining useful life prediction across operating conditions. Reliability Engineering and System Safety, 2021; 211: 107556.

[69] Do Van P, Barros A, Bérenguer C, Bouvard K, Brissaud F. Dynamic grouping maintenance with time limited opportunities. Reliability Engineering and System Safety, 2013; 120: 51–9.

[70] Mendes A A, Lorenzoni M W. Analysis and optimization of periodic inspection intervals in cold standby systems using Monte Carlo simulation. Journal of Manufacturing Systems, 2018; 49: 121–130.

[71] Zhang F, Shen J, Liao H, Ma Y. Optimal preventive maintenance policy for a system subject to two-phase imperfect inspections. Reliability Engineering and System Safety, 2021; 205: 107254.

[72] Marseguerra M, Zio E, Podofillini L. Condition-based maintenance optimization by means of genetic algorithms and Monte Carlo simulation. Reliability Engineering and System Safety, 2002; 77: 151–166.

[73] Dekker R, Wildeman RE, Schouten FAD. A review of multi-component maintenance models with economic dependence. Mathematical Methods of Operations Research, 1997; 45 (3): 411–435.

[74] Ding F, Tian Z. Opportunistic maintenance for wind farms considering multi-level imperfect maintenance thresholds. Renewable Energy, 2012; 45: 175–182.

[75] Zhang C, Gao W, Guo S, Li Y, Yang T. Opportunistic maintenance for wind turbines considering imperfect, reliability-based maintenance. Renewable Energy, 2016; 103: 606–612.

[76] Li M, Wang M, Kang J, Sun L, Jin P. An opportunistic maintenance strategy for offshore wind turbine system considering optimal maintenance intervals of subsystems. Ocean Engineering, 2020; 216: 108067.

[77] Ma Z, Wang D, Wu S, Teng W, Liu Y. An improved opportunistic group replacement maintenance strategy for wind turbines. Energy Science & Engineering, 2020; 8 (10): 3627–3637.

[78] Walter G, Flapper S D. Condition-based maintenance for complex systems based on current component status and Bayesian updating of component reliability. Reliability Engineering and System Safety, 2017; 168: 227–39.

[79] Shi Y, Zhu W, Xiang Y, Feng Q. Condition-based maintenance optimization for multi-component systems subject to a system reliability requirement. Reliability Engineering and System Safety, 2020; 202: 107042.

[80] Aizpurua J I, Catterson V M, Papadopoulos Y, Chiacchio F, D'Urso D. Supporting group maintenance through prognostics-enhanced dynamic dependability prediction. Reliability Engineering and System Safety, 2017; 168: 171–188.

[81] Xu J, Liang Z, Li Y, Wang K. Generalized condition-based maintenance optimization for multi-component systems considering stochastic dependency and imperfect maintenance. Reliability Engineering and System Safety, 2021; 211: 107592.

[82] Shafiee M, Finkelstein M, Bérenguer C. An opportunistic condition-based maintenance policy for offshore wind turbine blades subjected to degradation and environmental shocks. Reliability Engineering and System Safety, 2015; 142: 463–471.

[83] Pérez E, Ntaimo L, Ding Y. Multi-component wind turbine modeling and simulation for wind farm operations and maintenance. Simulation, 2015; 91(4): 360–382.

[84] Tian Z, Liao H. Condition based maintenance optimization for multi-component systems using proportional hazards model. Reliability Engineering and System Safety, 2011; 96: 581–589.

[85] Duan C, Makis V, Deng C. An integrated framework for health measures prediction and optimal maintenance policy for mechanical systems using a proportional hazards model. Mechanical Systems and Signal Processing, 2018; 111: 285–302.

[86] Zheng R, Chen B, Gu L. Condition-based maintenance with dynamic thresholds for a system using the proportional hazards model. Reliability Engineering and System Safety, 2020; 204: 107123.

[87] Wang H K, Huang H Z, Li Y F, Yang Y J. Condition-based maintenance with scheduling threshold and maintenance threshold. IEEE Transactions on Reliability, 2016; 65 (2): 513–524.

[88] Huynh K T, Grall A, Bérenguer C. Assessment of diagnostic and prognostic condition indices for efficient and robust maintenance decision-making of systems subject to stress corrosion cracking. Reliability Engineering and System Safety, 2017; 159: 237–254.

[89] Ghamlouch H, Fouladirad M, Grall A. The use of real option in condition-based maintenance scheduling for wind turbines with production and deterioration uncertainties. Reliability Engineering and System Safety, 2019; 188: 614–623.

[90] Ma X, Liu B, Yang L, Peng R, Zhang X. Reliability analysis and condition-based maintenance optimization for a warm standby cooling system. Reliability Engineering and System Safety, 2020; 193: 106588.

[91] Wang Y, Li X, Chen J, Liu Y. A condition-based maintenance policy for multi-component systems subject to stochastic and economic dependencies. Reliability Engineering and System Safety, 2022; 219: 108174.

[92] Zhou P, Yin P T. An opportunistic condition-based maintenance strategy for offshore wind farm based on predictive analytics. Renewable and Sustainable Energy Reviews, 2019; 109: 1–9.

[93] Pater I, Mitici M. Predictive maintenance for multi-component systems of repairables with remaining-useful-life prognostics and a limited stock of spare components. Reliability Engineering and System Safety, 2021; 214: 107761.

[94] Gui J, Sun Z, Wen Y, Tao D, Ye J. A review on generative adversarial networks: algorithms, theory, and applications. 2020, arXiv preprint arXiv:2001.06937.

[95] Olmschenk G, Zhu Z, Tang H. Generalizing semi-supervised generative adversarial networks to regression using feature contrasting. Computer Vision and Image Understanding, 2019; 186: 1–12.

[96] Rezagholiradeh M, Haidar M A. Reg-gan: Semi-supervised learning based on generative adversarial networks for regression. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, 2806–2810.

[97] Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A, Chen X. Improved techniques for training gans. In: Advances in Neural Information Processing Systems, 2016, 2234–2242.

[98] Peterson L E. K-nearest neighbor. Scholarpedia, 2009; 4 (2): 1883.

[99] Ruder S. An overview of multi-task learning in deep neural networks. 2017, arXiv:1706.05098.

[100] Ellefsen A L, Æsøy V, Ushakov S, Zhang H. A comprehensive survey of prognostics and health management based on deep learning for autonomous ships. IEEE Transactions on Reliability, 2019; 68 (2): 720–740.

[101] Wang B, Lei Y, Li N, Li N. A hybrid prognostics approach for estimating remaining useful life of rolling element bearings. IEEE Transactions on Reliability, 2020; 69 (1): 401–412.

[102] Fang X, Paynabar K, Gebraeel N. Multistream sensor fusion-based prognostics model for systems with single failure modes. Reliability Engineering & System Safety, 2017; 159: 322–331.

[103] Amini A, Schwarting W, Soleimany A, Rus D. Deep evidential regression. in: Proceedings of the 34th Conference on Neural Information Processing Systems, 2020.

[104] Kendall A, Gal Y. What uncertainties do we need in bayesian deep learning for computer vision? In Advances in neural information processing systems, 2017, 5574–5584.

[105] Sankararaman S. Significance, interpretation, and quantification of uncertainty in prognostics and remaining useful life prediction. Mechanical Systems and Signal Processing, 2015; 52–53: 228–247.

[106] Saxena A, Goebel K, Simon D, Eklund N. Damage propagation modeling for aircraft engine run-to-failure simulation. in 2008 international conference on prognostics and health management, IEEE, 2008, 1–9.

[107] Stevens B. EXAKT reduces failures at Canadian Kraft Mill, www.omdec.com, 2006.

[108] Cumming G, Fidler F, Vaux D L. Error bars in experimental biology. The Journal of Cell Biology, 2007; 177 (1): 7–11.

[109] Li N, Lei Y, Lin J, Ding S X. An improved exponential model for predicting remaining useful life of rolling element bearings, IEEE Transactions on Industrial Electronics, 2015; 62 (12): 7762–7773.

[110] Cao Y, Jia M, Ding P, Ding Y. Transfer learning for remaining useful life prediction of multi-conditions bearings based on bidirectional-GRU network, Measurement, 2021; 178: 109287.

[111] Quionero-Candela J, Sugiyama M, Schwaighofer A, Lawrence N D. Dataset shift in machine learning. The MIT Press, 2009.

[112] Byttner S, Rögnvaldsson T, Svensson M. Consensus self-organized models for fault detection (COSMO), Engineering Applications of Artificial Intelligence, 2011; 24 (5): 833–839.

[113] Ali J B, Chebel-Morello B, Saidi L, Malinowski S, Fnaiech F. Accurate bearing remaining useful life prediction based on Weibull distribution and artificial neural network, Mechanical Systems and Signal Processing, 2015; 56: 150–172.

[114] Yosinski J, Clune J, Bengio Y, Lipson H. How transferable are features in deep neural networks? In Advances in neural information processing systems, 2014, 3320–3328.

[115] Wang J, Chen Y, Feng W, Yu H, Huang M, Yang Q. Transfer learning with dynamic distribution adaptation. ACM Transactions on Intelligent Systems and Technology, 2020; 11 (1): 1–25.

[116] Gretton A, Borgwardt K M, Rasch M J, Schölkopf B, Smola A. A kernel two-sample test. The Journal of Machine Learning Research, 2012; 13: 723–773.

[117] Zadeh L A. Fuzzy Sets. Information and control, 1965; 8: 338–353.

[118] Wu D, Lawhern V J, Gordon S, Lance B J, Lin C. Driver drowsiness estimation from EEG signals using online weighted adaptation regularization for regression (OwARR). IEEE Transactions on Fuzzy Systems, 2016; 29: 1522–1535.

[119] Belkin M, Niyogi P, Sindhwani V. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. The Journal of Machine Learning Research, 2006; 7 (85): 2399−2434.

[120] McElreath R. Statistical rethinking: A Bayesian course with examples in R and Stan. CRC Press, 2016.

[121] Li N, Xu P, Lei Y, Cai X, Kong D. A self-data-driven method for remaining useful life prediction of wind turbines considering continuously varying speeds. Mechanical Systems and Signal Processing, 2022; 165: 108315.

[122] Morimoto J, Doya K. Reinforcement learning state estimator. Neural Computation, 2007; 19 (3): 730–756.

[123] Han M, Tian Y, Zhang L, Wang J, Pan W. H∞ model-free reinforcement learning with robust stability guarantee. 2020, arXiv:1911.02875.

[124] Hu L, Wu C, Pan W. Lyapunov-based reinforcement learning state estimator. 2021, arXiv: 2010.13529.

[125] Tian Y, Chao M A, Kulkarni C, Goebel K, Fink O. Real-time model calibration with deep reinforcement learning. Mechanical Systems and Signal Processing, 2022; 165: 108284.

[126] Chao M A, Kulkarni C, Goebel K, Fink O. Fusing physics-based and deep learning models for prognostics. Reliability Engineering and System Safety, 2022; 217: 107961.

[127] Borguet S J. Variations on the Kalman filter for enhanced performance monitoring of gas turbine engines. Ph.D. thesis, Université de Liège, 2012.

[128] Wager S, Wang S, Liang P. Dropout training as adaptive regularization. 2013, arXiv: 1307.1493.

[129] Mnih V, Kavukcuoglu K, Silver D, Rusu A A, Veness J, Bellemare M G, Graves A, Riedmiller M, Fidjeland A K, Ostrovski G, Petersen S, Beattie C, Sadik A, Antonoglou I, King H, Kumaran D, Wierstra D, Legg S, Hassabis D. Human-level control through deep reinforcement learning. Nature, 2015; 518 (7540): 529–533.

[130] Berkenkamp F, Turchetta M, Schoellig A, Krause A. Safe model based reinforcement learning with stability guarantees. in Advances in Neural Information Processing Systems, 2017, 908–918.

[131] Barenthin M, Wahlberg B, Hjalmarsson H, Barkhagen M. Data-driven methods for L2-gain estimation. IFAC Proceedings Volumes, 2009; 42 (10): 1597–1602.

[132] Chen Y, Liang X, Zuo M J. Sparse time series modeling of the baseline vibration from a gearbox under time-varying speed condition. Mechanical Systems and Signal Processing, 2019; 134: 106342.

[133] Chao M A, Kulkarni C, Goebel K, Fink O. Aircraft engine run-to-failure dataset under real flight conditions for prognostics and diagnostics. Data, 2021; 6 (1): 5.

[134] Yuan T, Qin S J. Root cause diagnosis of plant-wide oscillations using Granger causality. Journal of Process Control, 2014; 24 (2): 450–459.

[135] Zhang H, Borghesani P, Randall R B, Peng Z. A benchmark of measurement approaches to track the natural evolution of spall severity in rolling element bearings. Mechanical Systems and Signal Processing, 2022; 166: 108466.

[136] Irawan C A, Ouelhadj D, Jones D, Stålhane M, Sperstad I B. Optimization of maintenance routing and scheduling for offshore wind farms. European Journal of Operational Research, 2017; 256 (1): 76–89.

[137] Jafari L, Naderkhani F, Makis V. Joint optimization of maintenance policy and inspection interval for a multi-unit series system using proportional hazards model. Journal of the Operational Research Society, 2017; 69 (1): 36–48.

[138] Qiu Q, Cui L, Shen J, Yang L. Optimal maintenance policy considering maintenance errors for systems operating under performance-based contracts. Computers & Industrial Engineering, 2017; 112: 147–155.

[139] Wang J, Zhao X, Guo X. Optimizing wind turbine's maintenance policies under performance-based contract. Renewable Energy, 2019; 135: 626–634.

[140] Mo H, Sansavini G. Impact of aging and performance degradation on the operational costs of distributed generation systems. Renewable Energy, 2019; 143: 426–439.

[141] Hong H P, Zhou W, Zhang S, Ye W. Optimal condition-based maintenance decisions for systems with dependent stochastic degradation of components. Reliability Engineering and System Safety, 2014; 121: 276–288.

[142] Liu J, Wang H, Ji X. Energy efficiency evaluation of wind turbine based on AHP. In: Chinese Automation Congress (CAC), 2017, 4528–4531.

[143] WINDPOWER. https://www.thewindpower.net/, 2019.

[144] Wind turbine maintenance: components, strategies, and tools. https://blog.enerpac.com/wind-turbine-maintenance-components-strategies-and-tools/, 2021.

[145] Datla S V, Pandey M D. Estimation of life expectancy of wood poles in electrical distribution networks. Structural Safety, 2006; 28 (3): 304–319.

[146] Kealy T, Barrett M, Kearney D. How profitable are wind turbine projects? An empirical analysis of a 3.5 MW wind farm in Ireland. International Journal on Recent Technologies in Mechanical and Electrical Engineering, 2015; 2: 58–63.

[147] Martin-Tretton M, Reha M, Drunsic M, Keim M. Data collection for current US wind energy projects: Component costs, financing, operations, and maintenance. Contract, 2012; 303: 275–3000.

[148] Beauson J, Laurent A, Rudolph D P, Jensen J P. The complex end-of-life of wind turbine blades: A review of the European context. Renewable and Sustainable Energy Reviews, 2022; 155: 111847.

[149] Byrne R, Astolfi D, Castellani F, Hewitt N J. A study of wind turbine performance decline with age through operation data analysis. Energies, 2020; 13 (8): 2086.

[150] Papi F, Cappugi L, Salvadori S, Carnevale M, Bianchini A. Uncertainty quantification of the effects of blade damage on the actual energy production of modern wind turbines. Energies, 2020; 13 (15): 3785.

[151] Ahmad R, Kamaruddin S. An overview of time-based and condition-based maintenance in industrial application. Computers & Industrial Engineering, 2012; 63 (1): 135–149.

[152] Alhmoud L, Wang B. A review of the state-of-the-art in wind-energy reliability analysis. Renewable and Sustainable Energy Reviews, 2018; 81: 1643–1651.

[153] Tian Z, Zhang H. Wind farm predictive maintenance considering component level repairs and economic dependency. Renewable Energy, 2022; 192: 495–506.

[154] Tavner P. Offshore wind turbines: reliability, availability and maintenance. London: Institution of Engineering and Technology, 2012.

[155] Maples B, Saur G, Hand M, Pietermen R V D, Obdam T. Installation, operation, and maintenance strategies to reduce the cost of offshore wind energy. United States: National Renewable Energy Lab (NREL), 2013.

[156] Ren Z, Verma A S, Li Y, Teuwen JJE, Jiang Z. Offshore wind turbine operations and maintenance: A state-of-the-art review. Renewable and Sustainable Energy Reviews, 2021; 144: 110886.

[157] Hofmann M, Sperstad I B. Will 10 mw wind turbines bring down the operation and maintenance cost of offshore wind farms? Energy Procedia, 2014; 53: 231–8.

[158] Zhao X, Ren L. Focus on the development of offshore wind power in China: has the golden period come? Renewable Energy, 2015; 81: 644–657.

[159] Leitea GNP, Araújoa AM, Rosas PAC. Prognostic techniques applied to maintenance of wind turbines: a concise and specific review. Renewable and Sustainable Energy Reviews, 2018; 81: 1917–1925.

[160] Du Y, Zhou S, Jing X, Peng Y, Wu H, Kwok N. Damage detection techniques for wind turbine blades: A review. Mechanical Systems and Signal Processing, 2020; 141: 106445.

[161] Tian Z, Jin T, Wu B, Ding F. Condition based maintenance optimization for wind power generation systems under continuous monitoring. Renewable Energy, 2011; 36 (5): 1502–1509.

[162] Bakir I, Yildirim M, Ursavas E. An integrated optimization framework for multi-component predictive analytics in wind farm operations & maintenance. Renewable and Sustainable Energy Reviews, 2021; 138: 110639.

[163] Mérigaud A, Ringwood J V. Condition-based maintenance methods for marine renewable energy. Renewable and Sustainable Energy Reviews, 2016; 66: 53–78.

[164] Pinciroli L, Baraldi P, Ballabio G, Compare M, Zio E. Optimization of the operation and maintenance of renewable energy systems by deep reinforcement learning. Renewable Energy, 2022; 183: 752–763.

[165] Rezamand M, Kordestani M, Carriveau R, Ting D S, Orchard ME, Saif M. Critical wind turbine components prognostics: A comprehensive review. IEEE Transactions on Instrumentation and Measurement, 2020; 69 (12): 9306–9328.

[166] Wang Y, He R, Tian Z. Opportunistic condition-based maintenance optimization for electrical distribution systems. Reliability Engineering and System Safety, 2023; 236: 109261.

[167] Mrad A B, Delcroix V, Maalej M A, Piechowiak S, Abid M. Uncertain evidence in Bayesian networks: Presentation and comparison on a simple example. In International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, 2012, 39–48.

[168] Gebraeel N Z, Lawley M A, Li R, Ryan J K. Residual-life distributions from component degradation signals: A Bayesian approach. IIE Transactions, 2005; 37 (6): 543–557.

[169] Yildirim M, Gebraeel N Z, Sun X A. Integrated predictive analytics and optimization for opportunistic maintenance and operations in wind farms. IEEE Transactions on Power Systems, 2017; 32 (6): 4319–4328.

[170] Li M, Jiang X, Carroll J, Negenborn RR. A multi-objective maintenance strategy optimization framework for offshore wind farms considering uncertainty. Applied Energy, 2022; 321: 119284.

[171] Martin-Tretton M, Reha M, Drunsic M, Keim M. Data collection for current U.S. wind energy projects: Component costs, financing, operations, and maintenance; January 2011-September 2011. No. NREL/SR-5000-52707, National Renewable Energy Lab (NREL), 2012.

[172] Poore R, Walford C. Development of an operations and maintenance cost model to identify cost of energy savings for low wind speed turbines: July 2, 2004 – June 30, 2008. No. NREL/SR-500-40581, National Renewable Energy Lab (NREL), 2008.

[173] Badyda K, Dylik M. Analysis of the impact of wind on electricity prices based on selected European countries. Energy Procedia, 2017; 105: 55–61.

[174] Faulstich S, Hahn B, Tavner P J. Wind turbine downtime and its importance for offshore deployment. Wind energy, 2011; 14 (3): 327–337.

[175] He R, Tian Z, Zuo M J. A semi-supervised GAN method for RUL prediction using failure and suspension histories. Mechanical Systems and Signal Processing, 2022; 168: 108657.

[176] He R, Tian Z, Zuo M J. A transferable neural network method for remaining useful life prediction. Mechanical Systems and Signal Processing, 2023; 183: 109608.

[177] He R, Tian Z, Zuo M J. Machine prognostics under varying operating conditions based on state-space and neural network modeling. Mechanical Systems and Signal Processing, 2023; 182: 109608.

[178] He R, Tian Z, Wang Y, Zuo M J, Guo Z. Condition-based maintenance optimization for multi-component systems considering prognostic information and degraded working efficiency. Reliability Engineering & System Safety, 2023; 234: 109167.