Nonparamteric Bayesian HDP-HSMM: Modeling, Inference and Application to Remaining Useful Life Estimation

Yanwan Cao

BUSN 41916

December 12th 2025

ABSTRACT

This article focuses on the problem of finding correct number of hidden states in the hidden markov and hidden semi markov model, especially in the application of remaining useful life prediction in complex industrial systems. The goal is to build a prediction model framework that has both strong robustness and interpretability. Three types of methods are studied in this work: Hidden Markov Model, Hidden Semi-Markov Model, and Long Short Term Memory model. In particular, this article introduces the Hierarchical Dirichlet Process into the Hidden Semi-Markov Model, and constructs the Hierarchical Dirichlet Process-Hidden Semi-Markov Model together with its sampling algorithm. This model can automatically learn the segmentation and transition structure of hidden states under unsupervised conditions, which solves the problem of predefining the number of states in traditional Hidden Semi-Markov Moel. This kind of modeling approach combines flexibility and the ability to handle uncertainty, and is more suitable for complex industrial scenarios where the system states are not clear, samples are imbalanced, and devices are heterogeneous. Both simulated data and real datasets, including C-MAPSS and PHM08, are used in the experiments to evaluate the three types of models. The results show that the Hierarchical Dirichlet Process-Hidden Semi-Markov Model has better generalization ability under limited data conditions, can steadily capture the multi-stage degradation trend from health to failure, and shows better performance in expressing uncertainty. The main contribution of this article is the combination of Hierarchical Dirichlet Process-based unsupervised learning and Hidden Semi-Markov Model-based duration modeling for the remaining useful life prediction task. This expands the application of nonparametric Bayesian models in industrial forecasting and provides an interpretable modeling approach for future research.

Keywords: Remaining Useful Life Prediction; Hierarchical Dirichlet Process; Hidden Semi-Markov Model; Bayesian Nonparametrics

Table of Contents

Chapter 1 Introduction 1
1.1 Background 1
1.2 Motivation and Significance 1
Chapter 2 Literature Review 2
2.1 Remaining Useful Life 2
2.2 Statistical-Based Predictive Modeling 2
2.2.1 Hidden Markov Model 2
2.2.2 Hidden Semi-Markov Model 3
2.3 Deep Learning Model 6
Chapter 3 Modeling and Algorithm Design 7
3.1 Nonparametric Bayes Model 7
3.1.1 Stick-Breaking Process (GEM) 7
3.1.2 Dirichlet Process 8
3.1.3 Hierarchical Dirichlet Process 9
3.1.4 Sampling Algorithm 11
3.2 Deep Learning Modeling 15
Chapter 4 Simulation and Case Study 17
4.1 Simulation Data 17
4.2 C-MAPSS Case Study 20
4.2.1 Results from HDP-HSMM Model 21
4.2.2 Results from LSTM Model 26
4.3 PHM08 Case Study 28
4.3.1 Results from HDP-HSMM Model 29
4.3.2 Results From LSTM Model 31
Chapter 5 Conclusion and Future Work 33
5.1 Summary of the Results 33

5.2 Sensitivity Analysis 35
5.2.1 Influences by Each Parameter 35
5.2.2 Influences by distinct r and p 37
5.3 Limitations 38
5.3.1 Monotonic degradation path 38
5.3.2 Sampling Efficiency 38
5.4 Future Work 39
5.4.1 Accelerated sampling 39
5.4.2 Introducing the mechanism of covariate effects 40
References 41

Chapter 1 Introduction

1.1 Background

In the field of modern communication services, base stations, as key equipment in wireless communication networks, perform many critical functions such as signal transmission and reception, data transmission, spectrum management, connection, and switching management. However, base station equipment faces serious aging problems during long-term operation. Furthermore, the natural degradation process of electronic components due to continuous operation also causes irreversible damage to the base station. This damage intensifies over time, affecting equipment performance and leading to instability in network and data transmission.

The concept of remaining useful life (RUL) has been widely applied in industry. Monitoring and predicting RUL is becoming a core focus for various industries in achieving condition monitoring, early warning diagnostics, and intelligent decision-making.

1.2 Motivation and Significance

Although the problem of remaining useful life (RUL) has been widely applied and has achieved a certain degree of development, when facing practical problems, it is often affected by the equipment's data storage mechanism and related limiting factors. At the equipment level, these limitations can lead to various problems. Therefore, improving the accuracy of base station RUL prediction under limited data conditions becomes a critical technical challenge. Furthermore, because multiple base stations collaboratively provide limited data, determining the state of different base stations and their respective contributions to RUL prediction presents high-dimensional and heterogeneous challenges.

Chapter 2 Literature Review

2.1 Remaining Useful Life

Remaining Useful Life (RUL) refers to the time or operating cycles that a system or component can continue to function before reaching a predetermined failure criterion. RUL is a dynamic random variable that changes over time and is influenced by objective physical conditions, including external environmental factors and the mechanical and chemical changes of internal components.

The second stage is the remaining useful life prediction stage. Prediction is arguably the core task of life cycle management. This stage focuses on predicting the remaining operating time and service life of the equipment before any failure occurs, based on existing data. This provides a basis for subsequent maintenance work and optimization of resource allocation. The main goal of this stage is to achieve the most accurate predictions possible.

2.2 Statistical-Based Predictive Modeling

2.2.1 Hidden Markov Model

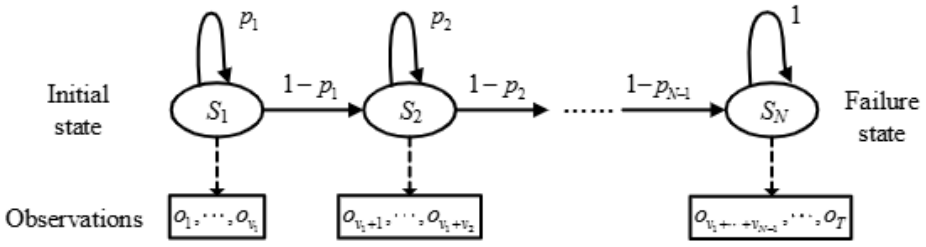

Figure 2.1 Modeling methods of HMM in RUL prediction

In hidden markov model, there are mainly 3 parameters:

$$\lambda = (A, B, \pi)$$

Table 2.1 Parameter definitions in the HMM model

Params	Math Definitions	Physical Meanings
States Set S	$S = s_1, s_2, \ldots, s_N$	Hidden States of the System N
Observations Set V	$V = v_1, v_2, \ldots, v_M$	Possible Observatiosn M 个
Initial Probability Distribution π	$\pi_i = P(q_1 = s_i)$	At $t = 1$ Probability of at State s_i
Probability Transition Matrix A	$a_{ij} = P\left(q_{t+1} = s_j \mid q_t = s_i\right)$	Probability from s_i to s_j
Observation Probability Matrix B	$b_j(k) = P\left(o_t = v_k \mid q_t = s_j\right)$	Probability of observing v_k at s_j

Suppose Observation has length T:

$$O = \{O_1, O_2, \ldots, O_T\}$$

Every O_t means the observation at time t, we expect to infer the hidden states behind the obersvations:

$$Q = \{q_1, q_2, \ldots, q_N\}$$

To infer the hidden states, we need to answer the following main 3 questions:

Table 2.2 Three core problems that HMM needs to address

Number	Questions
1	Given params $\lambda = (A, B, \pi)$, calculate the probability of observing O, i.e. $P(O\|\lambda)$?
2	Given observation O, find the most possible hidden state sequenec Q^*
3	Given observation, etimate optimal parameter $\lambda = (A, B, \pi)$, that maximize likelihood

Facing the three types of problems mentioned above, the current mainstream algorithms include the forward-backward algorithm, the Viterbi algorithm, and the Baum-Welch algorithm, the latter of which can be considered an EM-based learning method.

2.2.2 Hidden Semi-Markov Model

In a Hidden Markov Model (HMM), the true degradation state is assumed to evolve

according to a Markov process, but this state is not directly observable; only a sequence of observed data is available. The Hidden Semi-Markov Model (HSMM) is a generalization of the HMM, which explicitly models the state dwell time, replacing the default geometric distribution assumption in the HMM.

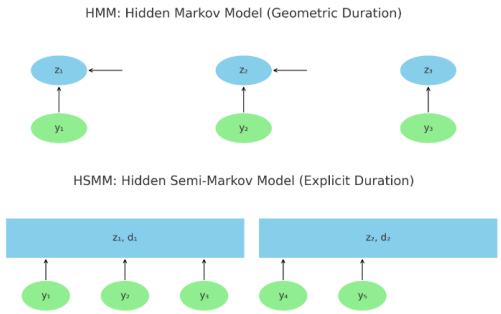

Figure 2.2 Comparison of latent variable durations in HMM and HSMM

Figure 3.4 illustrates the main differences between Hidden Markov Models (HMMs) and Hidden Semi-Markov Models (HSMMs). The difference lies in the method of modeling the duration of hidden states, where "duration" refers to the number of time steps the hidden state persists, not the actual time interval of observed data collection. In both HMMs and HSMMs, the observation sequence $O = O_1, O_2, \ldots, O_T$. It is generally assumed that the data is sampled at equal intervals, meaning each observation point corresponds to a unit time step. Hidden Markov Models (HMMs) have a fundamental assumption that the time interval for the system to transition from the current hidden state to the next state follows a geometric distribution. This is determined by the Markov property, which states that at any given time, the state transition depends only on the current state and is independent of the duration of the previous state. HMMs implicitly model the dwell time in each state as a geometric distribution, equivalent to having a fixed transition probability at each time step. However, in practical applications, this assumption is often invalid, as devices or systems generally remain in a certain state for a longer period, which does not conform to the "high-frequency short-stay" characteristic of the geometric distribution. To overcome this limitation, Hidden Semi-Markov Models (HSMMs) extend the HMM model by explicitly introducing a state

duration distribution. Specifically, HSMMs add a set of duration distribution parameters $D_j(d)$ to model the probability that the system remains in state j for d time steps. This improvement allows the model to more flexibly characterize the "phasic dwelling" behavior in real-world systems, improving the model's expressive power and fitting performance.

The fundamental assumptions of the hidden semi-Markov model differ significantly from those of the hidden Markov model, as explained earlier. When constructing this model and determining its parameters, the modeling process generally follows that of the hidden Markov model, but with the addition of parameters $D_j(d)$, to represent the duration distribution of the latent state variables.

$$P(O,Q,D \mid \lambda) = \pi_{q_1} D_{q_1}(d_1) \cdot \prod_{t=1}^{d_1} b_{q_1}(O_t) \cdot \prod_{k=2}^{K} a_{q_{k-1},q_k} D_{q_k}(d_k) \prod_{t=t_k}^{t_k+d_k-1} b_{q_k}(O_t) \Bigg]$$

By introducing a discrete hidden state structure, HMM/HSMM can approximate the system degradation path in scenarios where observations are discontinuous, states are difficult to measure directly, and the amount of sampled data is limited. This characteristic aligns with the practical features of communication base station scenarios where states cannot be directly observed and data is sparsely distributed. Furthermore, HMM/HSMM allows the model to automatically infer the degradation stages and their transition patterns based on the observation sequence, possessing strong fault tolerance and explanatory power.

However, this modeling framework faces a key challenge: how to determine and define the number of system states. The degradation states of communication base stations are difficult to classify into a finite and stable set of discrete categories. State classification is subjective and prone to human bias, and a fixed number of states limits the model's ability to represent complexity. To address this problem, subsequent research introduced non-parametric Bayesian methods such as the Dirichlet Process (DP) and Hierarchical Dirichlet Process (HDP), allowing the state space to adaptively expand during the modeling process, effectively mitigating the modeling errors caused by a fixed state structure.

2.3 Deep Learning Model

In the current environment of Industry 4.0, high-dimensional, non-linear, and noisy real-world monitoring data present significant challenges. With the continuous development of big data and artificial intelligence, deep learning models, with their robust end-to-end modeling capabilities and automatic feature extraction abilities, have become one of the mainstream trends in remaining useful life (RUL) prediction research in recent years. Data preprocessing, as the initial stage, aims to mitigate the interference of data noise on the results and standardize feature scales to improve the stability of the model training process. Common methods include Z-score normalization, dynamic differencing, and time-series windowing (S2S/S2P). The next stage involves health indicator construction, using feature transformation and fusion to generate indicators that reflect the underlying health status. These indicators should possess characteristics such as monotonicity, predictability, and consistent trends. Finally, the remaining useful life prediction stage involves selecting a suitable deep neural network model to map the health indicators or raw data obtained from the previous two stages to the RUL, providing an estimate of the remaining useful life. In addition to using original deep neural network structures, methods such as incorporating attention mechanisms, dropout, and adaptive optimizers can be used to improve the model's robustness and generalization ability.

Chapter 3 Modeling and Algorithm Design

3.1 Nonparametric Bayes Model

In the modeling process described earlier, we observed a situation where if the number of latent states is not properly matched to the true structure of the data, underfitting may occur (when the number of states is too small), or overfitting may occur (when the number of states is too large). In practical engineering scenarios, the number of latent states is often difficult to determine accurately in advance. We desire a model that can adaptively determine the number of states, allowing the data to drive the learning of the state structure. This modeling objective aligns well with the idea of the Dirichlet Process (DP) in non-parametric Bayesian methods. As a "distribution of distributions," DP can generate models with an infinite number of potential categories, allowing the model to dynamically adjust its complexity based on the amount of data. It automatically reduces the number of states when the data is limited and retains sufficient expressive power when the data is abundant.

To achieve the goal of adaptive expansion of the state space, the following sections will explain the basic principles and inference mechanisms of the Dirichlet Process, and analyze how to naturally integrate it into the modeling framework of the Hidden Semi-Markov Model, constructing a hierarchical Dirichlet Process Hidden Semi-Markov Model with state persistence modeling capabilities and automatic state number learning capabilities.。

3.1.1 Stick-Breaking Process (GEM)

Before introducing the Dirichlet process, let's discuss the stick-breaking process to illustrate the specific mechanism of data generation. Suppose we have a stick of length 1, and we want to break it into infinitely many segments, each segment representing the proportion or weight of a particular category.

$$\pi_k = \upsilon_k \prod_{l=1}^{k-1} 1 - \upsilon_l$$

υ_k satisfies $Beta(1,\alpha)$, $\alpha > 0$ is used to control the concentration level of the parameters, υ_1 represents the ratio of the position of the first break to the total length of the rod of length 1, $\upsilon_2$refers to the ratio of the length of the second break to the length of the remaining rod after the first break. By repeating this process, continuously breaking the remaining portion each time, we can ultimately obtain:

$$\sum_{k=1}^{\infty} \pi_k = 1$$

This ensures that it represents a concept of weighting, and also forms a probability distribution.

3.1.2 Dirichlet Process

The stick-breaking process is often associated with the Dirichlet process G.

$$G \sim DP(\alpha, H)$$

G：This is a type of discrete random distribution, constructed from a random measure generated by a Dirichlet Process. H：Base measure. α is the concentration parameter, with a value greater than 0. The concept of concentration means that the larger the value of α, the more weights will appear, and the resulting random distribution $G$the concentration parameter, with a value greater than 0. The concept of concentration means that the larger the value of α, the more weights will appear, and the resulting random distribution $\alpha = 0$ the random distribution G becomes a specific value, concentrating all the weights on a single data point.

For any collections of finite partition of sets $A_1, \ldots, A_k$, we have:

$$\big(G(A_1), \ldots, G(A_k)\big) \sim \text{Dirichlet}\big(\alpha G_0(A_1), \ldots, \alpha G_0(A_k)\big)$$

Which demonstrates that the Dirichlet process under a finite partition behaves as a Dirichlet distribution.

$$G = \sum_{k=1}^{\infty} \pi_k \, \delta_{\theta_k}, \quad \theta_k \sim H$$

This reflects Dirichlet's generative mechanism. Whenever a data point θ_k is generated, a weight π_k is generated according to the stick-breaking process. Here, π_k represents the probability of taking the value θ_k while the probability of not taking the value θ_k is $1-\pi_k$. The random distribution Ggenerated by the Dirichlet process can support an infinite number of potential clusters, but since most of the weight is concentrated on the first few terms, and subsequent weights tend towards zero, the number of clusters will converge, making it very suitable for classification tasks.

3.1.2.1 Inapplicability in HSMM Model

The Dirichlet process (DP) provides powerful structural adaptability for non-parametric modeling, allowing the model complexity to be adjusted based on the data without pre-setting the number of latent states. However, directly applying DP to hidden semi-Markov models (HSMMs) presents fundamental structural limitations. In the DP modeling framework, data points are treated as independent and identically distributed samples, and their classification is driven solely by the similarity between observed data. This allows DP to discover potential cluster structures through clustering, but it also means that DP cannot explicitly model the sequential dependencies between data points, and cannot represent the "probability structure of transitioning from one hidden state to another." In HSMMs, the core characteristic of the system lies precisely in the temporal evolution of hidden states, satisfying: (1) the dependence of the current state on future states is defined by the state transition probability matrix A; (2) each state maintains a specific duration before transitioning to the next state; (3) both temporal order information and the persistence and transitions between states jointly determine the generative structure of the entire model. However, the basic assumptions and generative mechanism of DP make it difficult to naturally define a state transition matrix. While it can tell us "which states exist," it cannot tell us "how states transition between each other." This makes DP more suitable for unordered clustering tasks, rather than modeling the evolution of hidden states with a clear temporal structure. Directly embedding DP into HSMMs would cause the model to lack the ability to describe the Markovian structure, contradicting the fundamental requirement of "state dependence" in HSMMs for tasks such as degradation modeling and equipment state prediction.

3.1.3 Hierarchical Dirichlet Process

To address the structural deficiencies mentioned above, we introduced the Hierarchical Dirichlet Process (HDP), a natural extension of the Dirichlet Process (DP). It allows a set of related distributions, such as the transition distributions of multiple states, to share a common global base distribution. That is, in HDP, each state j corresponds to a transition distribution π_j and these $\pi_j$are modeled as being sampled from a DP. All π_j share a global discrete distribution β constructed using a stick-breaking process. This allows for a clearly defined state transition structure even without pre-setting the number of states, enabling Markov modeling in an infinite state space.

By incorporating a duration modeling mechanism, HDP can be extended into HDP-HSMM, which allows for the automatic learning of the number of states, Markovian modeling of state transitions, and explicit modeling of state durations simultaneously. This approach offers strong expressive power and good interpretability, making it particularly suitable for degradation modeling and remaining useful life prediction tasks in complex time-series systems.

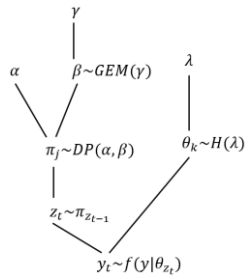

Figure 3.1 Hierarchical Dirichlet Process

γ is the concentration parameter of the GEM distribution, which controls the sparsity of the global state distribution β refers to the global state weight distribution obtained through the stick-breaking process, representing the probability of using the k-th state; α refers to the concentration parameter of the Dirichlet process corresponding to each state transition distribution π_j, where π_j is the distribution of transition probabilities to different states when in state j; z_t is the current state at time t, determined by the transition distribution z_{t-1}'s transition distribution $\pi_{z_{t-1}}$; λ is the hyperparameter of the observation distribution prior H; which controls the prior information of the distribution parameter θ_k; θ_k is the observation distribution parameter corresponding to the k-th state; y_t is the t-th observation, determined by the

parameterθ_{z_t} of the current state z_t.

3.1.3.1 Applicability in HSMM Model

Compared to the Dirichlet process, the Hierarchical Dirichlet Process (HDP) adds an extra layer of stick-breaking process to its structure. This process generates a globally shared state space β, and the transition distribution π_j for each state is constructed as a Dirichlet process with β as the base distribution, i.e., $\pi_j \sim \mathrm{DP}(\alpha, \beta)$. This structure allows all states to transition from the same infinite set of latent states. Thus, even without pre-defining the number of states, a clear state transition probability matrix can be defined, successfully solving the problem of traditional DP's inability to model state transitions. Simultaneously, it achieves the ability to construct a state transition matrix in an infinite state space. Therefore, the HDP structure is fundamental to building the HDP-HSMM.

Existing literature shows that hidden semi-Markov chains perform well in modeling collaborative data for remaining useful life prediction. However, the determination of the number of states has rarely been discussed. The hierarchical Dirichlet process (HDP) statistical model has been extensively discussed and applied in the context of hidden Markov models and hidden semi-Markov models, encompassing various fields such as speech recognition, medical time series analysis, and robot behavior analysis. However, research applying the hierarchical Dirichlet process to remaining useful life analysis is very rare. This paper will refine the research problem, focusing on the prediction performance of HDP-HSMM in remaining useful life prediction.

3.1.4 Sampling Algorithm

The posterior inference of the HDP-HSMM model involves multiple latent variables, such as the state sequence, transition distribution, duration distribution, and global stick-breaking weights. Since their joint posterior distribution is difficult to express analytically, an approximate inference method based on MCMC is used to sample these variables.

3.1.4.1 MCMC Sampling

Markov Chain Monte Carlo (MCMC) is a class of algorithms specifically designed

for sampling from complex probability distributions. Its basic idea is to construct a Markov chain $\{X_t\}_{t=1}^{\infty}$ such that the target distribution $\pi(x)$ is its stationary distribution. After the chain runs for a sufficiently long time, the distribution of its state sequence gradually approaches $\pi(x)$ and samples from the chain can be used to approximate the statistical properties of the target distribution. Unlike direct sampling methods, MCMC does not require normalization of the target distribution, making it particularly suitable for Bayesian inference problems where the posterior distribution is difficult to express analytically or has a complex structure. Common MCMC methods include: the Metropolis-Hastings algorithm, Gibbs sampling, slice sampling, and auxiliary variable methods widely used in non-parametric Bayesian modeling.

The core idea of MCMC is to construct a Markov chain X_t with the target distribution $\pi(x)$ as its stationary distribution, and then use the long-term behavior of this chain to sample from it and approximate the target distribution.

$$\lim_{t\to\infty} P(X_t = x) = \pi(x)$$

By using MCMC sampling, it's possible to generate samples corresponding to a specific distribution simply by providing its core terms. The HDP-HSMM model can then be evaluated by observing the predictive performance of the sampled data compared to the actual results.

3.1.4.2 HDP-HSMM Sampling

In this study, to perform joint sampling of the hidden state sequence, state transition probabilities, and duration distributions in the HDP-HSMM model, we employ various sampling strategies within the MCMC framework. Specifically, this includes using the Forward-Filtering Backward-Sampling method for state sequence sampling, Gibbs Sampling for updating local variables and hyperparameters, and combining stick-breaking construction with Beta/Gamma posterior updates to achieve iterative inference of the entire model.

Based on the structural properties of different variables in the model, we selected targeted sampling algorithms.

- For the hidden state sequence $\{z_t\}$ and the duration sequence $\{d_t\}$ the Forward-Filtering Backward-Sampling method is used for joint sampling. This

method relies on forward recursion and backward sampling, effectively avoiding sample degradation caused by path dependence.

- For the transition distribution π_j of each state and the global stick-breaking weights β since direct sampling from the posterior distribution is not possible, the Metropolis-Hastings algorithm is used for sampling. In the transition distribution, because the state space is a high-dimensional discrete distribution, traditional random walk MH methods often converge slowly. Therefore, an adaptive MH method is introduced, which dynamically adjusts the proposal covariance based on historical sampling information to improve sampling efficiency
- For the hyperparameters γ and αwe will use Gibbs sampling or slice sampling methods. In this specific structure, the conditional posterior distribution of γ generally exhibits a Gamma conjugate structure, allowing for explicit sampling, while α is updated using an auxiliary variable.
- In some continuous parameter updates, such as mean and covariance estimation in the case of Gaussian observation distributions, we consider introducing the MALA (Metropolis-adjusted Langevin Algorithm) sampling strategy, which utilizes gradient information to improve sampling efficiency in high-dimensional spaces.

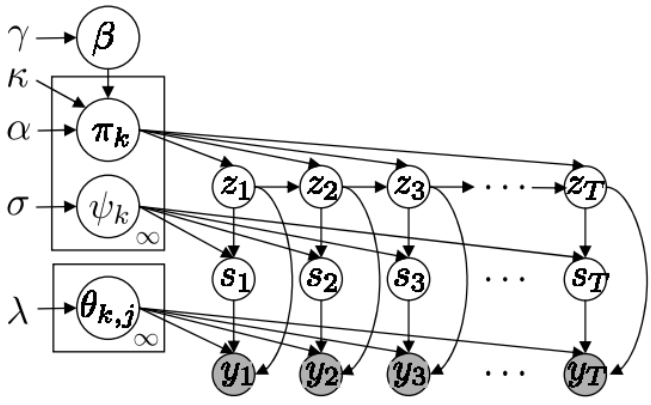

Figure 3.2 HDP-HSMM framework under the RUL prediction framework

This visually demonstrates the generation mechanisms and update processes of various parameters within the remaining useful life framework. The basic process can

be represented by the pseudocode algorithm.

Algorithm: HDP-HSMM posterior inference and RUL prediction

Input: Observation sequenc $\boldsymbol{y}_{1:T}$: Maximum Hidden state K, hyper-parameter (α, γ), prior parameter (a_α, b_α), $\left(a_\gamma, b_\gamma\right)$, $\{\theta_k\}, \{\lambda_k\}$, Gibbs iteration time S, failure state $\mathcal{Z}_F \subset 1, \ldots, K$

Output: $\alpha, \gamma, \beta, \{\pi_j\}, \{\theta_k\}, \{\lambda_k\}, \{z_t, d_t\}, \quad RUL_T^{(1)}, RUL_T^{(2)}, \ldots, RUL_T^{(S)}$

1：Initialize all the parameters with some BURN-IN procedure

2：**Step 1: Samplie all the hyper parameters**

3：**for** $\forall j$ **do**

4：Sampling $n_j \sim Beta(\alpha + 1, N_j)$

5：Calculate $m = \sum_j m_j, L = \sum_j log\,\eta_j, \pi = \frac{a_\alpha + m - 1}{(b_\alpha - L) + a_\alpha + m - 1}$

6：Sampling

$$\alpha \sim \pi \cdot Gamma(a_\alpha + m, b_\alpha - L) + (1 - \pi) \cdot Gamma(a_\alpha + m - 1, b_\alpha - L)$$

7：**End**

8：Samling $\eta \sim Beta(\gamma + 1, M)$

9：Calculate $\pi = \frac{a_\gamma + K - 1}{(b_\gamma - log\eta) + a_\gamma + K - 1}$

10：Sampling

$$\gamma \sim \pi \cdot Gamma\left(a_\gamma + K - 1, b_\gamma - log\eta\right) + (1 - \pi) \cdot Gamma\left(a_\gamma + K, b_\gamma - log\eta\right)$$

11：**Step 2: Sample** $\boldsymbol{\beta}, \{\boldsymbol{\pi_j}\}$

12：Sampling $\beta \sim \text{Dirichlet}(\gamma/K + m_1, \ldots, \gamma/K + m_K)$

13：**for** $\forall j$ **do**

14：Sampling $\pi_j \sim \text{Dirichlet}\left(\alpha \cdot \beta + N_j\right)$

15：**End**

16：**Step 3：Sampling** $\{\boldsymbol{z_t}, \boldsymbol{d_t}\}$

17：**for** t **do**

18： Sampling

$$z_t, d_t \sim p(z_t = k, d_t = d \mid y_{1:T}) \propto \pi_{z_{t-1},k} \cdot p(d_t) \cdot \prod_{s=t}^{t+d-1} p(y_s \mid \theta_k) \cdot \beta_{t+d}$$

19： **End**

20： **Step 4: Sampling** $\{\boldsymbol{\mu_k}, \boldsymbol{\sigma_k^2}, \boldsymbol{\lambda_k}\}$

21： **for** $k \leftarrow 1$ **to** S **do**

22： Sampling $\left(\mu_k, \sigma_k^2 \mid y_t : z_t = k\right) \sim$ Normal-Inverse-Gamma

23： Sampling $\lambda_k \sim \text{Gamma}(a + \sum d_t,\ b + \sum d_t)$

24： **End**

25： **Step 5： generate RUL predictive distribution**

26： **for** $s \leftarrow 1$ **to** S **do**

27： Record $z_{T+1}, d_{T+1}, z_{T+1+d_{T+1}}, d_{T+2}$, until $z_{t^*} \in \mathcal{Z}_F$

28： $\text{RUL}_T^{(s)} = t^* - T$

29： **End**

After sampling using Algorithm as stated above, we obtain a set of posterior samples for RUL, from which various estimators can be derived. Table 3..

$$\text{RUL}_T^{(1)}, \text{RUL}_T^{(2)}, \ldots, \text{RUL}_T^{(S)}$$

Table 3.1 RUL Estimation in HDP-HSMM

Estimate	Formula
Point Estimation	$\widehat{RUL}_T = \frac{1}{S}\sum_{s=1}^{S} \text{RUL}_T^{(s)}$
95% Credible Interval	$\text{RUL}_T^{(0.025)}, \text{RUL}_T^{(0.975)}$
Density Function	$p(\text{RUL}_T \mid y_{1:T}) \approx \frac{1}{S}\sum_{s=1}^{S} \delta_{\text{RUL}_T^{(s)}}$

3.2 Deep Learning Modeling

Deep learning exhibits advantages in processing large-scale, non-linear, and high-dimensional time series data, making it particularly suitable for industrial equipment health prediction tasks. This paper attempts to model base station time series data using

Long Short-Term Memory (LSTM) networks to predict their remaining useful life (RUL).

Compared to traditional recurrent neural networks, LSTM networks have a more powerful ability to model temporal dependencies, effectively preventing the vanishing gradient problem. They are suitable for handling gradual degradation trends in long sequences. Compared to structures like Transformers, LSTM is more computationally efficient in tasks with limited data or high real-time requirements. For the typical point-by-point observation and non-periodic degradation data in this task, LSTM can more naturally fit its evolutionary patterns.

The model employs a standard single-layer LSTM architecture, with the input being a time series of length T, $x_1, x_2, \dots, x_T$ where each $x_t \in \mathbb{R}^d$ represents a multi-dimensional observation at a specific time point. The hidden state of the LSTM network is denoted as $h_t \in \mathbb{R}^H$, We select only the hidden state at the last time step, h_T as the comprehensive health representation of the device at that moment, and then input it into a fully connected layer for remaining useful life regression prediction, where $W \in \mathbb{R}^{1\times H}$, $b \in \mathbb{R}$ are the weight parameters of the regression layer.

$$\widehat{RUL} = W h_T + b$$

Given that the degraded data may contain noise and outliers, the L1 loss function is chosen as the training objective. Compared to the L2 loss function, it is more robust to outliers.

$$\mathscr{L} = \frac{1}{N}\sum_{i=1}^{N}\left|RUL_i - \widehat{RUL}_i\right|$$

Where RUL_i represents the true Remaining Useful Life (RUL) of the i-th sample, and $\widehat{RUL}_i$ is the model's predicted value.

Chapter 4 Simulation and Case Study

4.1 Simulation Data

Regarding the data generation mechanism: First, four hidden states are predefined: True 0, 1, 2, and 3. For each state, a duration distribution and an observation distribution are set. These states are then generated sequentially, with each state lasting for 10 time steps. During these 10 steps, observation values are sampled from the corresponding Gaussian distribution. The entire observation sequence and the true state sequence are thus constructed. These observation data are then fed into the HDP-HSMM model. This model, without knowing the true number of states, automatically learns the state partitioning, duration distribution, and the state-observation correspondence. Finally, the state sequence inferred by the model is compared with the true state sequence.

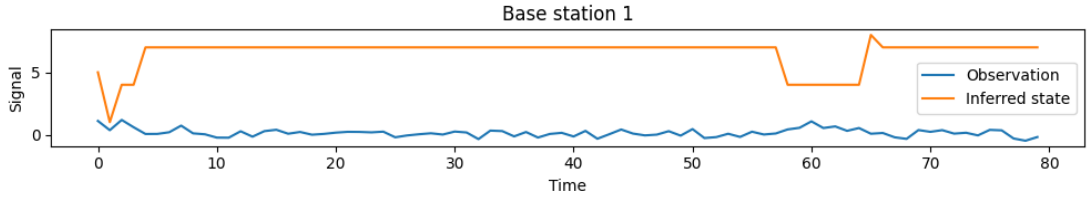

Figure 4.1 Comparison of observed and potential states at base station 1

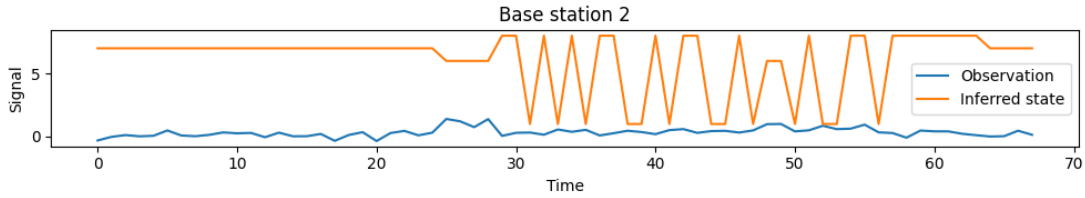

Figure 4.2 Comparison of observed and potential states at base station 2

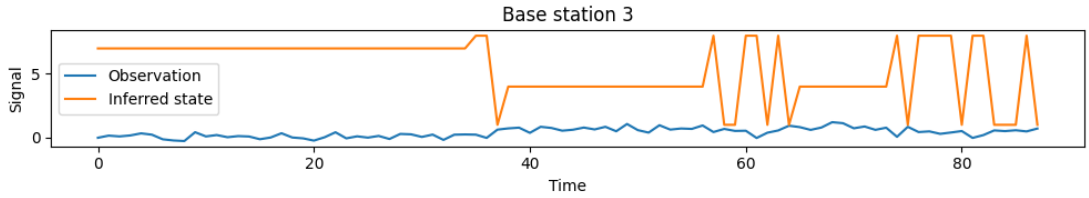

Figure 4.3 Comparison of observed and potential states at base station 3

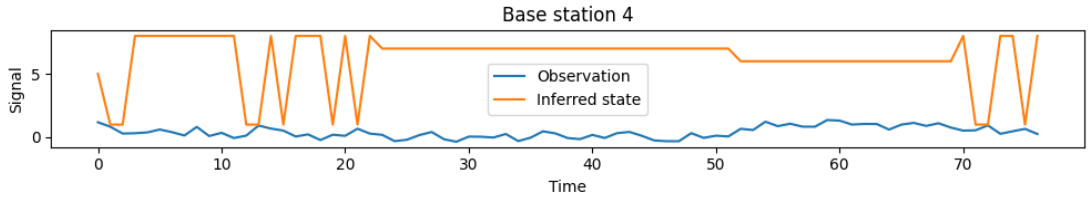

Figure 4.4 Comparison of observed and potential states at base station 4

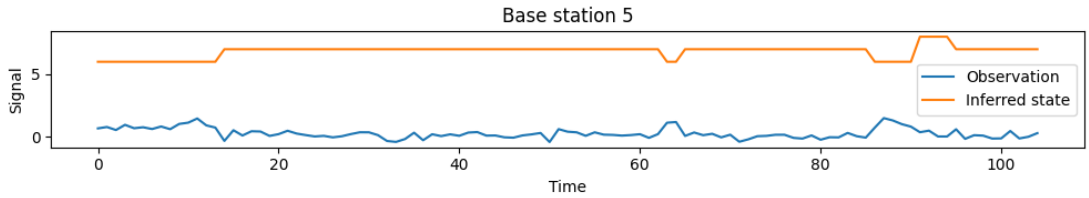

Figure 4.5 Comparison of observed and potential states at base station 5

Based on the inference results from multiple base stations, the HDP-HSMM model demonstrates excellent temporal structure modeling capabilities and sensitivity in state recognition. This model effectively captures abrupt changes in observed signals. For example, in Base station 3 (Figure 4.3), a prominent change in the observed signal occurs between time steps 35 and 40, and the inferred hidden state also undergoes a synchronous transition, demonstrating its rapid response capability to degradation processes or sudden events. Furthermore, in Base stations 1 (Figure 4.1) and 5 (Figure 4.5), the model maintains consistent state discrimination over extended periods. For instance, in Base station 5, the state remains stable from time step 10 to 60. This indicates that the model fully utilizes the duration modeling characteristics of HSMM, avoiding the frequent state switching problems common in traditional HMMs.

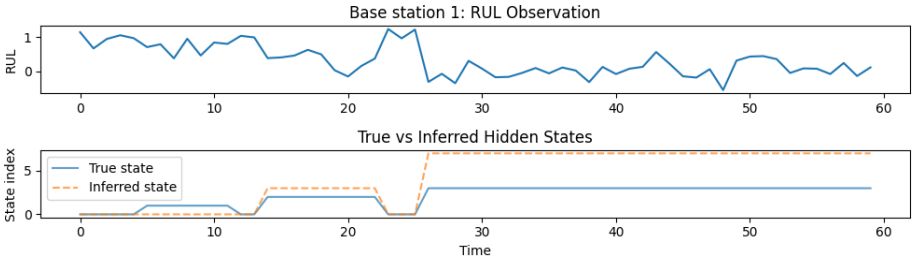

Figure 4.6 Comparison of actual and potential states of base station 1

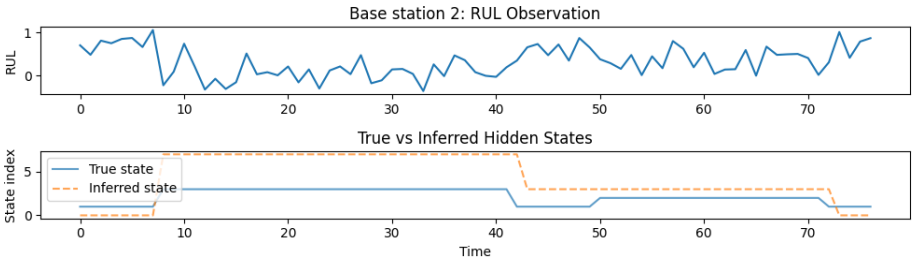

Figure 4.7 Comparison of actual and potential states of base station 2

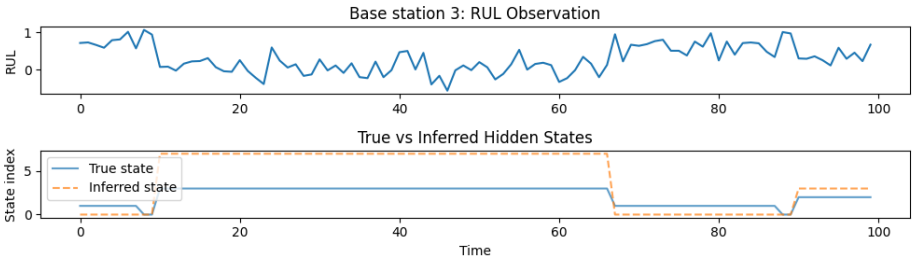

Figure 4.8 Comparison of actual and potential states of base station 3

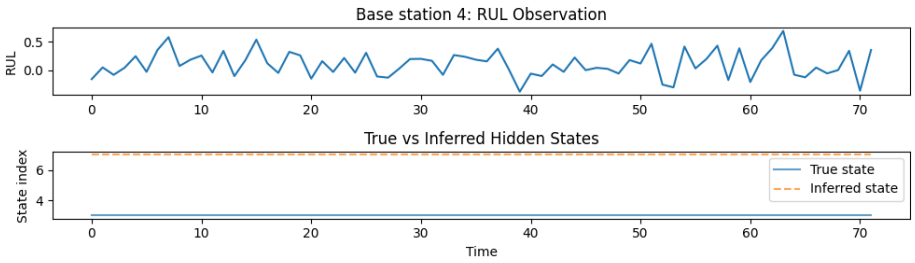

Figure 4.9 Comparison of actual and potential states of base station 4

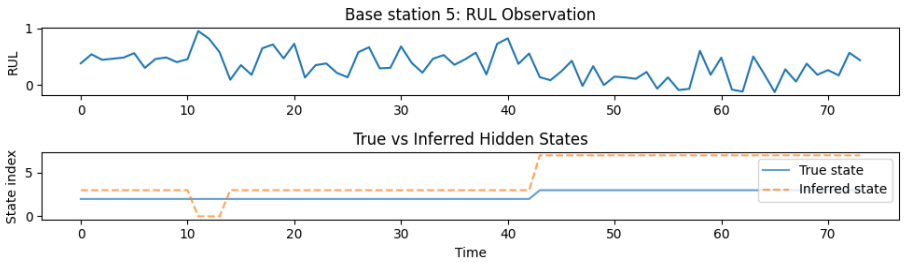

Figure 4.10 Comparison of actual and potential states of base station 5

By analyzing the simulation results from multiple base stations, it was found that HDP-HSMM performs well in fitting the real state transition trends. The hidden state sequence inferred by the model is consistent with the real state in terms of overall structural characteristics. In Base station 2 (Figure 4.7) and 3 (Figure 4.8), the model accurately captured the key state transition moments in the middle and later stages, and their timing is highly similar to the real state changes. In Base station 5 (Figure 4.10), the transition points identified by the model almost perfectly match the real labels, which confirms its high accuracy in state dynamics identification.

4.2 C-MAPSS Case Study

The CMAPSS simulation dataset, used for aircraft engine health monitoring and remaining useful life prediction, was released by the NASA Ames Research Center in the United States. This high-quality dataset utilizes real physical modeling and simulation tools to simulate the performance degradation process of aircraft engines under different operating conditions. It has wide applications in research areas such as predictive maintenance, fault detection, and state estimation.

Table 4.1 C-MAPSS Data Description

Dataset	# training trajectories	# test trajectories	# working conditions	# failure modes
FD001	100	100	1	1
FD002	260	259	6	1

FD003	100	100	1	2
FD004	248	249	6	2
Total time taken	single condition + Single fault	multiple conditions + Single fault	single condition + Multiple faults	multiple conditions + Multiple faults

4.2.1 Results from HDP-HSMM Model

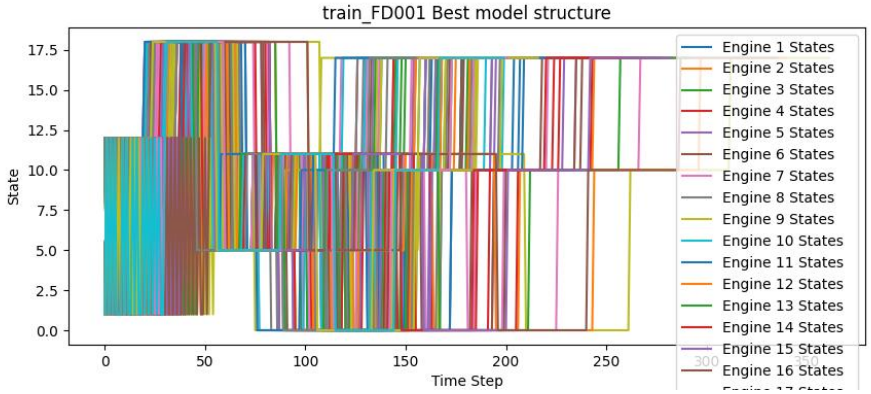

Figure 4.11 Comparison of potential states of different engines

In this experiment, we first performed HDP-HSMM modeling using the raw data to analyze the applicability of the model to this dataset. The raw sensor data exhibits typical multi-stage degradation characteristics, covering the initial stable operation, the gradual deterioration in the intermediate stage, and the drastic fluctuations before failure. This complex temporal structure places high demands on state modeling. Traditional HMMs, given their inherent assumption that state durations follow a geometric distribution, tend to switch states frequently, leading to severe "state fragmentation" in the inference results, failing to accurately reflect the equipment's operating cycle. HSMM, on the other hand, explicitly incorporates duration distributions into the modeling process, allowing states to persist for a certain period in a manner closer to the real physical process before transitioning. This avoids problems of artificial segmentation or misinterpretation of short-term states. By modeling the state dwell time,

HSMM can more naturally capture the phased evolution path of the system operation, resulting in a more coherent state sequence with stronger engineering interpretability.

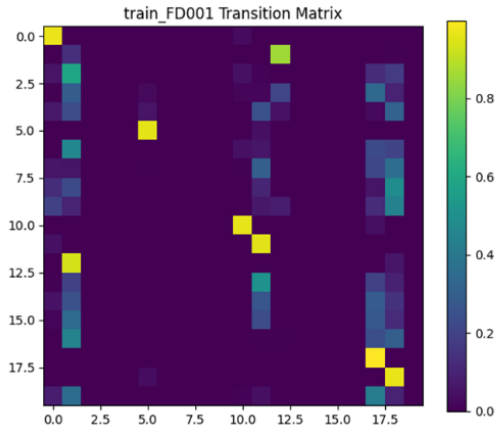

Figure 4.12 HDP-HSMM transition matrix

In the CMAPSS FD001 dataset, the state transition matrix learned by the HDP-HSMM model exhibits highly structured characteristics, powerfully demonstrating the model's advantages in complex time series modeling. Specifically, multiple prominent high-intensity regions are observed near the diagonal, indicating that the model effectively identifies the main path structure of the equipment's operating state evolution over time. This means that states mostly transition in the order of "state i to state i + 1," reflecting the typical process of equipment transitioning from healthy to degraded and finally to failure. This gradual transition pattern aligns with the real-world engineering logic of the CMAPSS data, where the equipment remains in a relatively stable operating phase for most of the time, then enters a slow degradation period, and finally fails.

The transition matrix as a whole exhibits significant sparsity. In Figure 4.14, except for a few states, most other positions are dark, indicating that the transition probability between most states is extremely low. The model automatically learns the few existing transition paths, assigning near-zero transition probabilities to non-physical jump states. This sparse structure improves the model's interpretability and robustness, avoiding arbitrary jumps and overfitting problems that may occur in traditional HSMMs. Some states, such as states 0, 12, and 18, show strong self-loop probabilities on the diagonal. This means that the model successfully captures the behavioral characteristics of the equipment potentially remaining in these stages for extended periods. This "state

persistence" often corresponds to the stable operating phase or slow degradation phase of the equipment, achieved through explicitly modeling duration distributions rather than simply state transitions. It is under the support of this modeling mechanism that HDP-HSMM effectively avoids frequent state transitions, better reflecting the actual operating conditions of industrial systems.

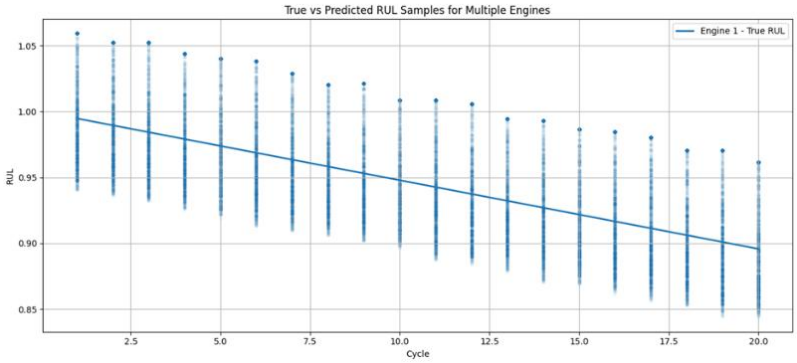

Figure 4.13 RUL prediction distribution generated from unfiltered data

Although this experiment has not yet introduced artificial masking points, directly using complete observational data for multi-engine sampling is still meaningful. On the one hand, this setup can be used to evaluate the model's baseline performance under complete information conditions, providing a reference standard for subsequent masking experiments. On the other hand, the prediction results on complete data can help us observe the model's fitting ability and stability at different degradation stages.

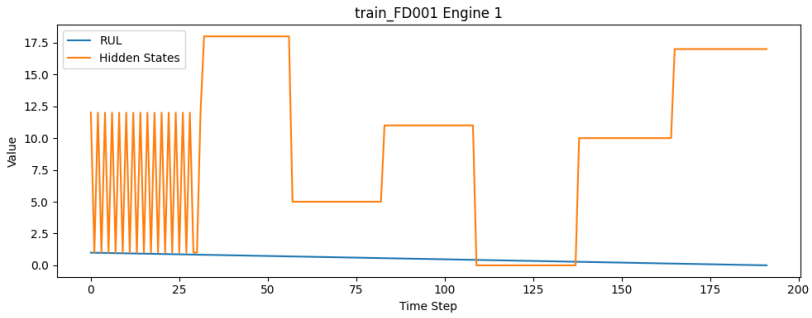

Figure 4.14 Potential state convergence figure generated without data masking

In the CMAPSS FD001 dataset, the state transition matrix learned by HDP-HSMM exhibits highly structured characteristics and good physical interpretability. Multiple high-probability transition regions are observed along the diagonal, indicating that the

model successfully captures the sequential changes between states, specifically the main path of gradual transition from a healthy state to a degraded state. This step-by-step progression aligns perfectly with the "slow degradation—gradual failure" engineering logic of industrial equipment operation. The matrix is highly sparse overall, with significant transitions only occurring between certain states. The model effectively filters out abrupt transitions that do not conform to the physical behavior of the equipment, making the state structure more robust and reliable. Some key states, such as states 0, 12, and 18, exhibit high self-loop probabilities, reflecting the model's accurate identification of stable or slowly changing phases of the equipment. This self-looping characteristic fully demonstrates the advantage of HDP-HSMM in explicitly modeling state durations, avoiding the frequent state switching problems of traditional HMMs. The model effectively reconstructs the phased evolution patterns in the equipment's operating life cycle and provides a clear and reliable structural basis for subsequent fault diagnosis, life prediction, and health management strategy development.

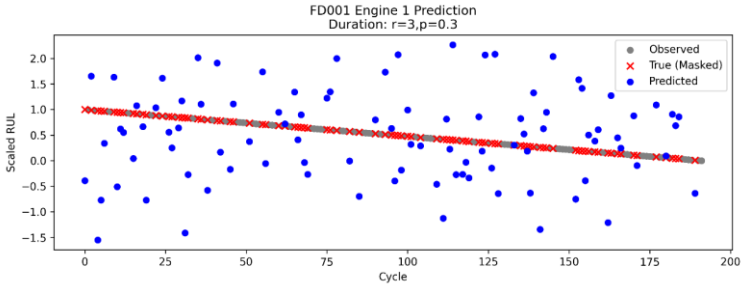

Figure 4.15 RUL prediction distribution generated under obscured data conditions

In Figure 4.15, the red crosses represent the masked true RUL values, the gray points are the observable true samples, and the blue points are the predicted samples generated by the model based on partial observational information. It can be clearly seen that the distribution of the predicted points covers the entire degradation range from high RUL to low RUL, indicating that the model has the ability to perform sampling and prediction throughout the entire life cycle and also exhibits good sample distribution breadth. The distribution range of the blue points shows that the model can still make reasonable inferences about the RUL state space even under high masking rates, demonstrating a certain degree of broad inference capability. However, due to the lack of a clear trend structure, further fitting and evaluation are needed to verify whether it

can accurately reconstruct the equipment degradation trajectory.

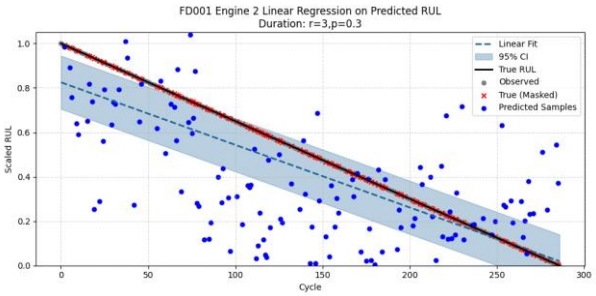

Figure 4.16 Linear regression for predicting RUL under censored data conditions

In Figure 4.16, the blue dashed line represents the linear fitting result of the predicted points, the black solid line represents the scaled true RUL curve, and the light blue shaded area represents the 95% confidence interval. Observing the overall trend, it can be seen that the fitted curve closely matches the true curve. This indicates that the model preserves the monotonic structure of the degradation path during the sampling process, and can accurately reconstruct the life cycle trend of the equipment even under incomplete information. Although there are some deviations in individual prediction points, the main trend of the fitting remains well-concentrated, and the confidence interval shows good convergence. This fully demonstrates the robustness and predictive stability of the model. The predicted points fluctuate around the true curve, indicating that the model does not simply output a single mean estimate, but possesses a certain variance modeling capability, reflecting the prediction uncertainty.

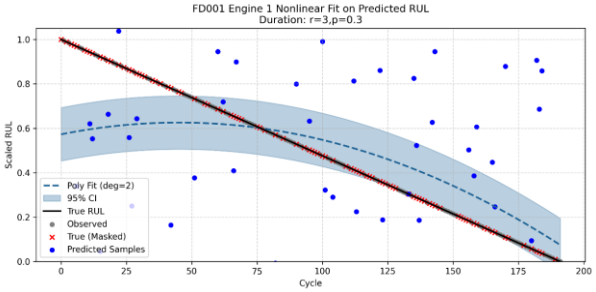

Figure 4.17 Nonlinear fitting and prediction of RUL under censored data

Using a quadratic polynomial for nonlinear fitting on the same set of prediction points, Figure 4.17 shows that even with a high coverage rate, the model can still capture

the nonlinear structure in the actual degradation process. In the later stages of the lifecycle, the predicted fitting curve shows a significant accelerated decline. This "nonlinear acceleration" is a typical characteristic of many industrial devices approaching failure. The confidence interval gradually increases over time, indicating that the model's predictions are more stable in the early stages, while the uncertainty of the predictions naturally increases as the equipment approaches failure, which is highly consistent with the risk evolution characteristics in practical applications. HDP-HSMM provides rich sampling predictions and possesses the ability to capture both monotonic and nonlinear trends, demonstrating excellent adaptability and interpretability in equipment life modeling.

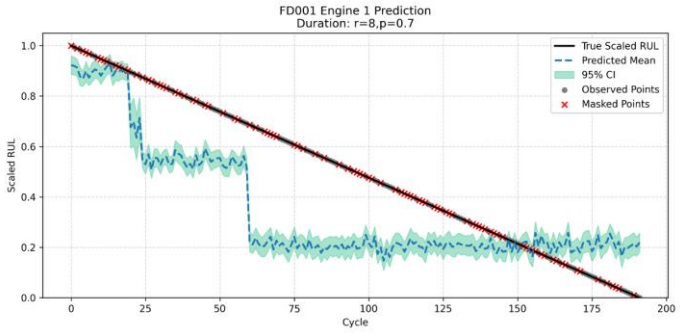

Figure 4.18 Potential state prediction under occluded data

Figure 4.18 shows the step-like prediction curve, which models and predicts the posterior mean of the latent states of the hidden data points. It can be clearly observed that this curve, with its small-amplitude jaggedness and large-amplitude step-like structure, exhibits a distinct piecewise structure, indicating the existence of three underlying health states behind the data. This structure is learned by the model from the overall data spontaneously, without requiring manually set thresholds or pre-defined state intervals, demonstrating the advantages of Bayesian latent state modeling in uncovering system degradation mechanisms.

4.2.2 Results from LSTM Model

This study selected LSTM over other deep learning methods because the equipment degradation process inherently exhibits time-dependent and phased characteristics. LSTM excels at capturing contextual relationships in long-term time series, and

compared to ordinary feedforward neural networks and convolutional neural networks, LSTM can effectively retain the influence of historical states on current predictions, preventing long-term dependency information from being forgotten in the sequence. This makes it more suitable for modeling the nonlinear degradation trends and lifespan changes of equipment during operation.

During the training phase, the model uses Mean Absolute Error (MAE) as the primary loss function and employs the Adam optimizer for efficient optimization. A comprehensive evaluation of the model's performance is conducted using a combination of metrics including MAE, Mean Squared Error (MSE), and fitted curves. This approach demonstrates strong sequence modeling capabilities, maintaining prediction stability while also depicting the dynamic characteristics of the degradation process, providing a high-accuracy deep learning solution for industrial equipment health management and lifespan prediction.

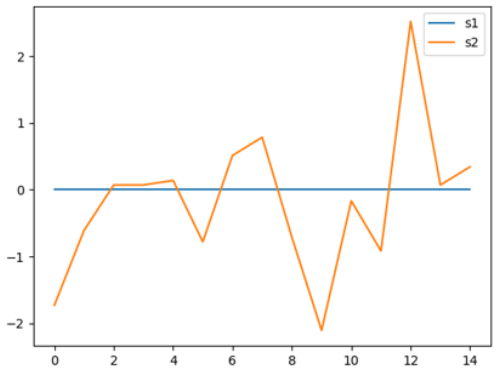

Figure 4.19 Comparison of data characteristics between s1 and s2

As shown in Figure 4.19, the normalized output results of the two sensors s1 and s2 within the same time window are significantly different. The values of sensor s1 change very smoothly, almost approaching a constant, with very small fluctuations. It does not contain effective time-series information in the current time period and does not show any prominent degradation trend. Sensor s2, on the other hand, exhibits stronger volatility and trend. Its values fluctuate significantly and include local maxima and minima. This dynamic change may correspond to changes in the equipment's operating state, offering higher representational capacity and predictive value. This result suggests that when using LSTM for data learning in the data modeling process,

sensor data with a clear temporal structure, like s2, can provide crucial support for RUL prediction. Redundant features like s1, which show almost no fluctuation, may be automatically assigned lower weights during model training or even weakened in hidden layer activation. If there are many redundant sensors, feature selection mechanisms or attention weighting modules can be introduced to increase attention to high-information features before training or within the model structure, thereby improving the model's convergence speed and generalization performance.

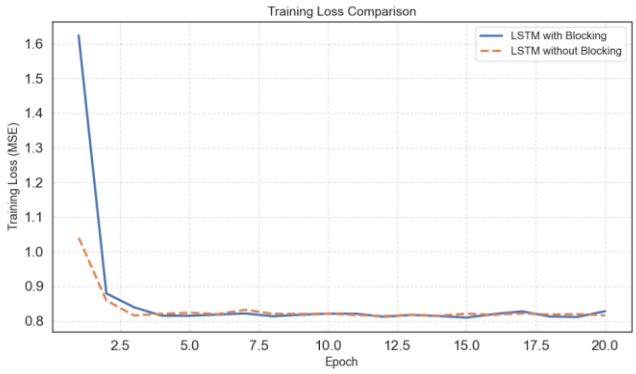

Figure 4.20 Comparison of training loss function convergence with and without blocks

To explore the impact of training strategies on the performance of deep temporal models, this study compared two LSTM training methods: one using a standard end-to-end training process, and the other introducing a "blocking" mechanism, which divides long sequences into several block-like subsequences to update model parameters in stages. From the trend of mean squared error with respect to epochs during the training process (Figure 4.20), it can be seen that the model with blocking exhibits faster convergence in the initial stages, with a more significant decrease in loss within the first three epochs. This mechanism can mitigate problems such as vanishing or exploding gradients and improve the efficiency of parameter updates.

4.3 PHM08 Case Study

The PHM08 dataset, released by NASA, contains data on commercial jet engines. It consists of multiple multivariate time series, each corresponding to an independent engine operating cycle, simulating the entire process from normal operation to

degradation and failure. The data is divided into a training set (train.txt), a test set (test.txt), and a final test set (final_test.txt). Each row records the observed data of a specific engine at a specific time, including the engine number, operating cycle number, operating condition parameters, and multiple sensor readings. Table 4.2 provides a general overview.

Table 4.2 PHM08 Data Description

Number	Variable Name	Meaning
1	unit_number	Engine number
2	time_in_cycles	The number of cycles at the time of the current observation.
3	operational_setting_1	Operating condition setting parameter 1
4	operational_setting_2	Operating condition setting parameter 2
5	operational_setting_3	Operating condition setting parameter 3
6-26	sensor_measurement_1 to sensor_measurement_21	Readings from a total of 21 sensors.

4.3.1 Results from HDP-HSMM Model

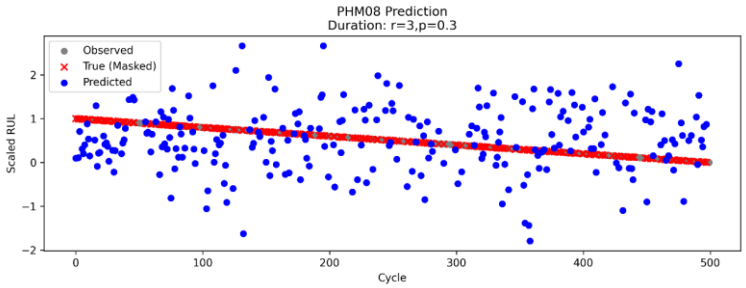

Figure 4.21 RUL prediction distribution generated under obscured data conditions

In Figure 4.21, the red crosses represent the masked true RUL values, the gray dots are the observable true data, and the blue scatter points are the predicted samples generated by the model. It can be observed that the predicted samples are distributed across the entire RUL range, covering the entire degradation phase from healthy equipment to imminent failure. This indicates that the model possesses strong temporal

structure awareness. The wide distribution of blue points at different RUL levels also shows that the model has a certain breadth of reasoning ability in the sample space. However, given the lack of a very prominent trend structure in the original scatter plot, further fitting analysis is needed to determine whether the model has truly captured the dynamic laws of the degradation process.

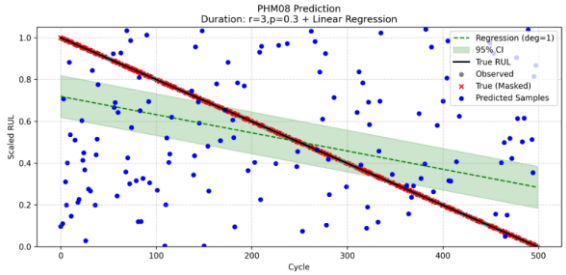

Figure 4.22 Linear regression-based RUL prediction under data masking conditions

Figure 4.22 shows the linear fitting results for the predicted samples. The results indicate that the overall trend closely matches the true RUL curve, meaning that the model preserves the monotonic structure of degradation at the sample generation level and possesses strong life cycle modeling capabilities. Although there are some outlier prediction points, the main trend closely follows the true degradation trajectory, and the 95% confidence interval converges well, which confirms the model's robustness and stability. Furthermore, the prediction points fluctuate above and below the true curve, indicating that the model has some variance estimation capability in its output, rather than simply outputting a single mean value.

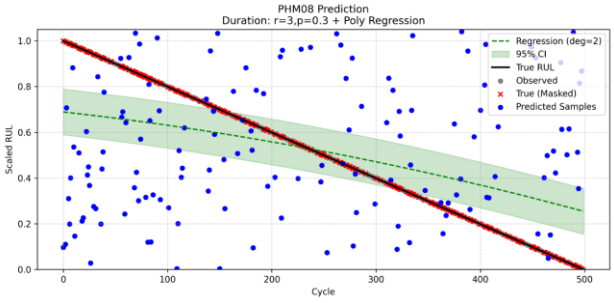

Figure 4.23 Nonlinear fitting for RUL prediction under obscured data conditions

Figure 4.23 uses a quadratic polynomial to fit the predicted samples, verifying the model's ability to capture nonlinear degradation trends. The fitted curve performs better

than linear fitting in the later stages, more closely approximating the accelerated decline phase of the true RUL. The model can identify the characteristic that the equipment degradation rate gradually increases over time. It is worth noting that the confidence interval gradually expands in the later stages of prediction, reflecting the model's natural response to uncertainty risk when the equipment is about to fail. This dynamic uncertainty modeling capability is crucial in practical industrial applications. Overall, HDP-HSMM possesses the ability to generate representative samples and accurately reconstruct degradation trends through sample fitting, demonstrating excellent structural modeling and predictive interpretability.

4.3.2 Results From LSTM Model

By observing Figure 4.24, it can be seen that the model trained on the PHM08 dataset exhibits a relatively flat loss curve throughout the training period. This means that the model converges relatively early in the learning process, but the overall decrease in error is relatively limited. The training loss initially decreases slightly and then remains almost constant at around 180, while the validation loss also remains stable at around 116 for a long time. Both curves show relatively small fluctuations. This phenomenon indicates that although the model has reached a stable training state, it has not adequately learned the underlying complex degradation patterns in the data, failing to further reduce the error.

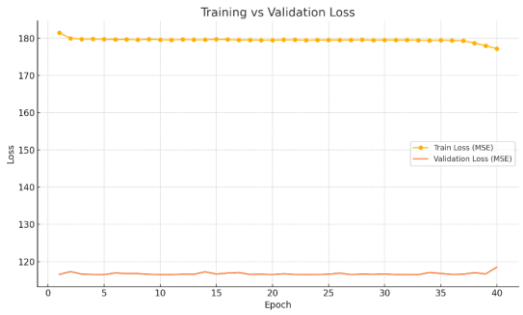

Figure 4.24 Comparison of Training and Validation Loss Functions

Figure 4.24 shows that the model training appears to be in a stable state. However, there is still room for improvement in its fitting ability and potential. Various methods can be employed in the future, such as increasing the model depth, introducing attention mechanisms or gating mechanisms, and strengthening feature selection and fusion, to

improve the model's ability to represent and predict complex degradation processes.

Chapter 5 Conclusion and Future Work

5.1 Summary of the Results

In the four datasets in C-MAPCSSand PHM08,

Table 5.1 MAPCSS - FD001 (3.35MB, Small, single-state degradation)

Model	Estimation	Mean Absolute Error (MAE)	Std. of Error
HSMM	Point Estimation	0.1294	0.1616
HDP-HSMM	Predictive Distribution	0.2018	0.2463
LSTM	Point Estimation	0.7315	0.8429

Table 5.2 MAPCSS - FD002 (8.66MB, Multi-modal, high complexity)

Model	Estimation	Mean Absolute Error (MAE)	Std. of Error
HSMM	Point Estimation	0.3256	0.4014
HDP-HSMM	Predictive Distribution	0.3019	0.3615
LSTM	Point Estimation	0.7224	0.8348

Table 5.3 MAPCSS - FD003 (4.01MB, Few operating conditions, slow degradation)

Model	Estimation	Mean Absolute Error (MAE)	Std. of Error
HSMM	Point Estimation	0.1185	0.1489
HDP-HSMM	Predictive Distribution	0.2950	0.3537
LSTM	Point Estimation	0.7751	0.8691

Table 5.4 MAPCSS - FD004 (9.87MB, High failure rate + multiple operating conditions)

Model	Estimation	Mean Absolute Error (MAE)	Std. of Error
HSMM	Point Estimation	0.3234	0.4015
HDP-HSMM	Predictive Distribution	0.2971	0.3554

LSTM	Point Estimation	7642	0.8632

Table 5.5 PHM08 (7.35MB, Height sensor nonlinearity + occlusion)

Model	Estimation	Mean Absolute Error (MAE)	Std. of Error
HSMM	Point Estimation	0.0608	0.0770
HDP-HSMM	Predictive Distribution	0.0388	0.0459
LSTM	Point Estimation	0.2245	0.2815

In terms of estimation accuracy, HDP-HSMM and HSMM performed similarly overall. However, in tasks with larger datasets, such as CMAPSS's FD002 (Table 5.2), FD004 (Table 5.4), and PHM08 (Table 5.5) datasets, HDP-HSMM demonstrated stronger modeling capabilities and higher prediction accuracy. This indicates that when dealing with equipment data with complex state structures and longer lifespans, HDP-HSMM's non-parametric modeling and duration modeling capabilities can effectively improve the model's fit to the true state evolution process. In cases with smaller datasets, such as FD001 (Table 5.1) and FD003 (Table 5.3), HSMM performed better, possibly due to its relatively simpler model structure, which makes it easier to converge with insufficient data. In comparison, the LSTM model showed generally lower prediction accuracy and weaker fitting capabilities across all tasks. It failed to adequately capture long-term dependencies in tasks with high coverage rates or non-linear degradation structures.

Table 5.6 C-MAPSS Running Time

CMAPSS	HSMM	HDP-HSMM	LSTM
FD001	2.26s	96.93s	2.41s
FD002	3.00s	102.27s	4.23s
FD003	2.61s	109.57s	2.47s
FD004	3.47s	100.78s	5.38s

In terms of runtime, HDP-HSMM is significantly slower than the other two methods, requiring longer training and inference times. However, its runtime is less sensitive to data size, exhibiting relatively stable computational requirements across different datasets. The HSMM and LSTM models, on the other hand, show

approximately linear increases in runtime with larger data volumes, their efficiency being more directly affected by the amount of data.

This presents a clear trade-off: HDP-HSMM excels in identifying potential hidden state structures and modeling system degradation mechanisms, performing particularly well when the number of states is unknown or the lifecycle structure is complex, but at the cost of higher computational expense. HSMM and LSTM models offer advantages in computational efficiency, making them suitable for industrial scenarios requiring faster response times or dealing with larger datasets, but they are relatively limited in their ability to express the intrinsic evolutionary patterns of the system. Balancing modeling capabilities and operational efficiency is a key consideration in practical deployment, requiring a comprehensive assessment based on task requirements.

5.2 Sensitivity Analysis

5.2.1 Influences by Each Parameter

In the field of non-parametric Bayesian methods, a question of great interest to many researchers is whether the prior distribution of the parameters has a stronger influence on the posterior distribution than the sampled data. If so, is the prior distribution correct? If not, how can we determine this? Therefore, Section 5.2.1 will discuss whether and how the prior distribution influences the posterior distribution by examining posterior distributions generated from different datasets.

Table 5.7 Parameter settings (Version 1) compared to other scenarios

	Version1	Version2	Version3	Version4	Version5
μ_0	1	**10**	1	1	1
σ_0	100	100	**0.01**	100	100
κ_0	0.25	0.25	0.25	**2500**	0.25
ν_0	10	10	10	10	**1000**

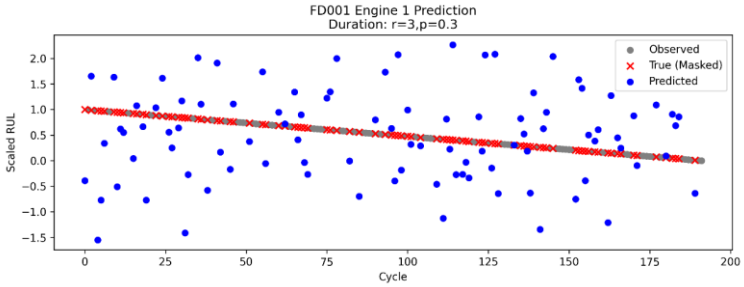

Figure 5.1 Sampling under Version 1 conditions

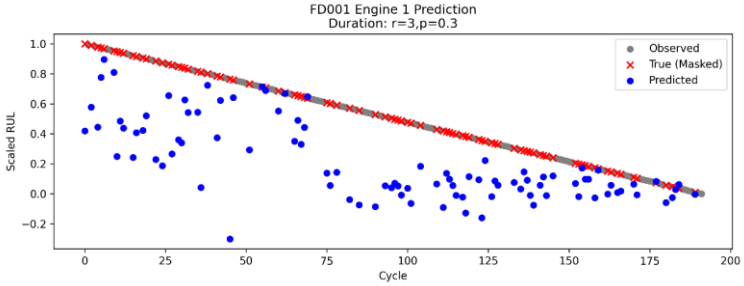

Figure 5.2 Sampling under Version 2 conditions

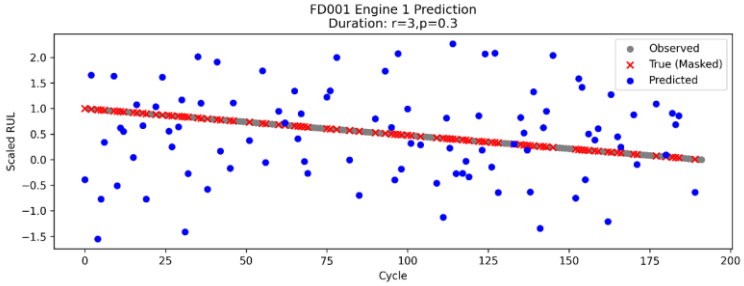

Figure 5.3 Sampling under Version 3 conditions

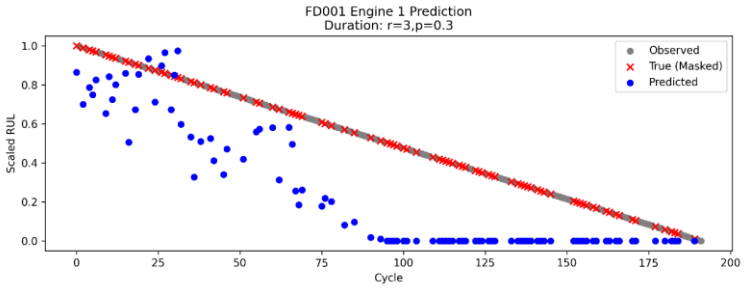

Figure 5.4 Sampling under Version 4 conditions

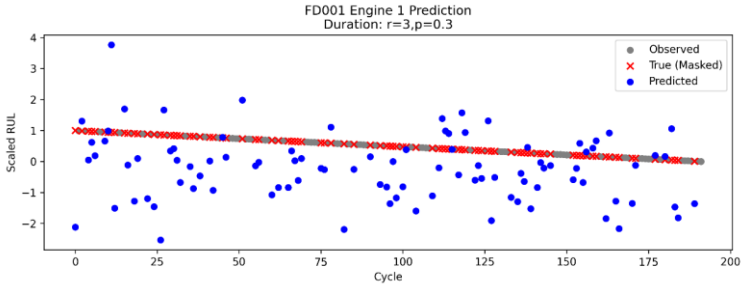

Figure 5.5 Sampling under Version 5 conditions

μ_0 serves as the prior center for the state mean. If it is set to an extreme value far from the true distribution of the data, it will significantly shift the model's interpretation of the observed data, causing a drift in the generated samples.

$\kappa_0$controls the model's confidence in this mean. A larger value indicates that the model is more "committed" to the prior, reducing the model's ability to learn from the data during training.

σ_0^2 and ν_0 together determine the prior strength of the observed distribution variance. Experimental results show that if σ_0 is set too small or ν_0 is too large, the model becomes overly conservative when modeling the uncertainty of the observed values, leading to "overconfident" predictions.

In practice, we should find a reasonable range for the parameter priors through multiple trials.

5.2.2 Influences by distinct r and p

r and p in the HDP-HSMM model refer to the fact that the duration of the latent states satisfies:

$$D_j(t) \sim \text{Neg} - \text{Bin}(r, p)$$

Therefore, we attempted to explore whether different time settings would affect the HDP-HSMM model.

Table 5.8 Sampling prediction results obtained with different r and p values.

r	***p***	MAE	RMSE
3	0.3	0.3019	0.3615

r	*p*	MAE	RMSE
20	0.1	0.3087	0.3697
5	0.5	0.3221	0.3845
8	0.7	0.3171	0.3777
1	0.9	0.3174	0.3778

Table shows the adjustments made to the duration distribution parameters r and p in the model. Although these parameters are necessary components for defining the negative binomial distribution within the modeling structure, their impact on the final prediction results is relatively limited. Regardless of the parameter settings, the model's output shows a high degree of consistency, both in terms of accuracy metrics and fitting trends. This means that the predictive performance of the HDP-HSMM is largely driven by the structure of the observed data itself, rather than relying on specific prior duration hyperparameter settings.

5.3 Limitations

5.3.1 Monotonic degradation path

In some monotonically degrading scenarios, such as irreversible processes like crack growth, the transition state of the latent variable is preset to move from one state to the immediately adjacent state. However, in the HDP-HSMM model, the latent states are typically inferred by reading all the data, without incorporating the information that "the current state can only transition to the next state." Therefore, although the transition probability matrix shows that the model can identify this underlying pattern in most cases, there is still a small probability of transitioning to other states. A solution to this problem is to reject the sample if it transitions to another state during sampling. However, the quantification of this rejection probability, its causes, parameter adjustment, and the discussion of its impact are not included in the scope of this paper. This is a promising direction for future research. This would not only improve the model's interpretability and predictive stability but also expand its application boundaries in practical engineering fields.

5.3.2 Sampling Efficiency

In this paper, the sampling method for the HDP-HSMM model primarily uses the MCMC algorithm. Traditional MH and Gibbs algorithms are less efficient, and the loss due to the initial burn-in period is significant, especially in high-dimensional state spaces. This is likely the reason for the data presented in Table 5.6. Furthermore, the code implementation in this paper is mainly based on Python; using a lower-level language like CPython might result in higher efficiency. However, in terms of model comparison, if all three models—HSMM, HDP-HSMM, and LSTM—were implemented using Cython, the conclusions regarding the comparison between the models should remain unchanged. Therefore, we look forward to implementing the HDP-HSMM model using Cython in the future to provide a more efficient interface for other research, experiments, and applications.

5.4 Future Work

5.4.1 Accelerated sampling

Variational Inference (VI), a statistical method that has become increasingly popular in recent decades, aims to approximate the true posterior distribution by iteratively finding a more tractable family of distributions (measured by Kullback-Leibler divergence). Compared to traditional MCMC methods, VI offers higher computational efficiency in large-scale data and complex models, making it particularly suitable for high-dimensional models or applications requiring extensive computation.

HDP-HSMM model, we want to calculate the joint posterior probability:

$$p\left(\beta, \pi_j, z_{1:T}, d_{1:T}, \theta, \omega \mid x_{1:T}\right)$$

Since this posterior distribution is difficult to compute directly, we approximate it by introducing a variational distribution $q(\cdot)$ with the goal of maximizing the Evidence Lower Bound (ELBO):

$$\log p(x_{1:T}) \geq \mathbb{E}_q \left[\log \frac{p(x_{1:T}, \text{latent})}{q(\text{latent})} \right] = \text{ELBO}$$

Introducing variational inference (VI) into the HDP-HSMM framework is an important way to improve its practical usability. Variational inference can effectively

accelerate the sampling process and is suitable for online learning or systems that need to quickly adapt to new data.

Furthermore, existing literature has explored variational modeling methods for HDP-HMM and HDP-HSMM, such as using mean-field approximation combined with stick-breaking representation, and performing joint variational inference on the state transition matrix, duration distribution, and observation parameters. These methods improve the convergence rate and give the model stronger scalability in big data scenarios.

5.4.2 Introducing the mechanism of covariate effects

Currently, the HDP-HSMM model is primarily used for unsupervised learning, and in data instance validation, only RUL (Remaining Useful Life) data is used. Other data, such as sensor selection and physical characteristics like temperature and humidity, are not considered. In the future, by introducing covariates into the HDP-HSMM model, analogous to the selection of numerous data features in regression analysis or deep learning, this model can incorporate more information and make more accurate predictions.

References

[1] Bunks, C., McCarthy, D., & Al-Ani, T. (2000). 36-Condition-based maintenance of machines using Hidden Markov Models. *Mechanical Systems and Signal Processing - MECH SYST SIGNAL PROCESS*, *14*, 597–612. https://doi.org/10.1006/mssp.2000.1309

[2] Dong, M., & He, D. (2007a). 2-Hidden semi-Markov model-based methodology for multi-sensor equipment health diagnosis and prognosis. *European Journal of Operational Research*, *178*, 858–878. https://doi.org/10.1016/j.ejor.2006.01.041

[3] Gebraeel, N., Lawley, M., Li, R., & Ryan, J. (2005). 4-Residual-life distribution from component degradation signals: A Bayesian approach. *Iie Transactions*, *37*, 543–557. https://doi.org/10.1080/07408170590929018

[4] S. Zheng, K. Ristovski, A. Farahat, & C. Gupta. (2017). 6-Long Short-Term Memory Network for Remaining Useful Life estimation. *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, 88–95. https://doi.org/10.1109/ICPHM.2017.7998311

[5] Arcieri, G., Hoelzl, C., Schwery, O., Straub, D., Papakonstantinou, K. G., & Chatzi, E. (2023). 14-Bridging POMDPs and Bayesian decision making for robust maintenance planning under model uncertainty: An application to railway systems. *Reliability Engineering & System Safety*, *239*, 109496. https://doi.org/10.1016/j.ress.2023.109496

[6] Si, X.-S., Wang, W., Hu, C.-H., & Zhou, D.-H. (2011). 24-Remaining useful life estimation – A review on the statistical data driven approaches. *European Journal of Operational Research*, *213*(1), 1–14. https://doi.org/10.1016/j.ejor.2010.11.018

[7] Dong, M., & He, D. (2007b). 32-A segmental hidden semi-Markov model (HSMM)-based diagnostics and prognostics framework and methodology. *Mechanical Systems and Signal Processing*, *21*, 2248–2266. https://doi.org/10.1016/j.ymssp.2006.10.001

[8] Lawless, J. F. (2003). *Statistical models and methods for lifetime data* (2nd ed.). Wiley. https://doi.org/10.1002/9781118033005

[9] Baysse, C., Bihannic, D., Gégout-Petit, A., Prenat, M., & Saracco, J. (2012). *36-Detection of a degraded operating mode of optronic equipment using Hidden Markov Model.* https://arxiv.org/abs/1212.2358

[10] Dong, M., & He, D. (2007b). 32-A segmental hidden semi-Markov model (HSMM)-based diagnostics and prognostics framework and methodology. *Mechanical Systems and Signal Processing*, *21*, 2248–2266. https://doi.org/10.1016/j.ymssp.2006.10.001

[11] Gomes, G., Lopes, S., Carrijo Polonio Araujo, D., Flauzino, R. A., Pinto, M., &

Alves, M. (2024). 38-Wind Turbine Remaining Useful Life Prediction Using Small Dataset and Machine Learning Techniques. *Journal of Control, Automation and Ele ctrical Systems*, *35*. https://doi.org/10.1007/s40313-024-01076-y

[12] Melo, A., Câmara, M. M., & Pinto, J. C. (2024). 39-Data-Driven Process Monitori ng and Fault Diagnosis: A Comprehensive Survey. *Processes*, *12*(2). https://doi.org/ 10.3390/pr12020251

[13] M. Wu, Q. Ye, J. Mu, Z. Fu, & Y. Han. (2023). 40-Remaining Useful Life Predi ction via a Data-Driven Deep Learning Fusion Model-CALAP. *IEEE Access*, *11*, 1 12085–112096. https://doi.org/10.1109/ACCESS.2023.3322733

[14] Ferreira, C., & Gonçalves, G. (2022). 42-Remaining Useful Life prediction and cha llenges: A literature review on the use of Machine Learning Methods. *Journal of Manufacturing Systems*, *63*, 550–562. https://doi.org/10.1016/j.jmsy.2022.05.010

[15] Chen, Z., Xia, T., & Pan, E. (2017). *43-Remaining Useful Life Estimation Based o n a Segmental Hidden Markov Model With Continuous Observations*. https://doi.org/ 10.1115/MSEC2017-2765

[16] Sethuraman, J. (1994). 45-A Constructive Definition of the Dirichlet Prior. *Statistic a Sinica*, *4*, 639–650. https://www3.stat.sinica.edu.tw/statistica/j4n2/j4n216/j4n216.htm

[17] Johnson, M., & Willsky, A. (2012). 46-Bayesian Nonparametric Hidden Semi-Mark ov Models. *Journal of Machine Learning Research*, *14*. https://arxiv.org/abs/1203.13 65

[18] Haario, H., Saksman, E., & Tamminen, J. (2001). 47-An Adaptive Metropolis Algo rithm. *Bernoulli*, *7*. https://doi.org/10.2307/3318737

[19] Teh, Y., Jordan, M., Beal, M., & Blei, D. (2006). 48-Hierarchical Dirichlet Process es. *Machine Learning*, 1–30. https://doi.org/10.1198/016214506000000302

[20] Li, Y., Schofield, E., & Gönen, M. (2019). 50-A tutorial on Dirichlet process mixt ure modeling. *Journal of Mathematical Psychology*, *91*, 128–144. https://doi.org/10.1 016/j.jmp.2019.04.004

[21] Aonan Zhang, San Gultekin, & John Paisley. (2016). 51-Stochastic Variational Infer ence for the HDP-HMM. Arthur Gretton & Christian C. Robert, *Proceedings of th e 19th International Conference on Artificial Intelligence and Statistics* (P 800–808). PMLR. https://proceedings.mlr.press/v51/zhang16a.html

[22] Johnson, M. J., & Willsky, A. S. (2014). 52-Stochastic variational inference for Ba yesian time series models. *31st International Conference on Machine Learning, IC ML 2014*, *5*, 3872–3880. https://proceedings.mlr.press/v32/johnson14.html