Uncertainty Quantification and Prognostics using Bayesian Statistics and Machine Learning

by

Jie Chen

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved March 2022 by the
Graduate Supervisory Committee:

Yongming Liu, Chair
Aditi Chattopadhyay
Marc Mignolet
Yi Ren
Hao Yan

ARIZONA STATE UNIVERSITY
May 2022

ABSTRACT

Uncertainty quantification is critical for engineering design and analysis. Determining appropriate ways of dealing with uncertainties has been a constant challenge in engineering. Statistical methods provide a powerful aid to describe and understand uncertainties. This work focuses on applying Bayesian methods and machine learning in uncertainty quantification and prognostics among all the statistical methods. This study focuses on the mechanical properties of materials, both static and fatigue, the main engineering field on which this study focuses. This work can be summarized in the following items: First, maintaining the safety of vintage pipelines requires accurately estimating the strength. The objective is to predict the reliability-based strength using nondestructive multimodality surface information. Bayesian model averaging (BMA) is implemented for fusing multimodality non-destructive testing results for gas pipeline strength estimation. Several incremental improvements are proposed in the algorithm implementation. Second, the objective is to develop a statistical uncertainty quantification method for fatigue stress-life (S-N) curves with sparse data.

Hierarchical Bayesian data augmentation (HBDA) is proposed to integrate hierarchical Bayesian modeling (HBM) and Bayesian data augmentation (BDA) to deal with sparse data problems for fatigue S-N curves. The third objective is to develop a physics-guided machine learning model to overcome limitations in parametric regression models and classical machine learning models for fatigue data analysis. A Probabilistic Physics-guided Neural Network (PPgNN) is proposed for probabilistic fatigue S-N curve estimation. This model is further developed for missing data and arbitrary output

distribution problems. Fourth, multi-fidelity modeling combines the advantages of low- and high-fidelity models to achieve a required accuracy at a reasonable computation cost. The fourth objective is to develop a neural network approach for multi-fidelity modeling by learning the correlation between low- and high-fidelity models. Finally, conclusions are drawn, and future work is outlined based on the current study.

ACKNOWLEDGMENTS

I would like to dedicate my sincere gratitude and appreciation to my advisor, Professor Yongming Liu, for his mentorship, support, and encouragement throughout my Ph.D. at Arizona State University. I would like to express my gratefulness to my committee members, Professor Aditi Chattopadhyay, Professor Marc Mignolet, Professor Yi Ren, and Professor Hao Yan, for their continuous help and valuable discussions. I would like to thank my colleagues and friends in my research group. Especially, thanks are owed to my parents and brother for their endless love.

TABLE OF CONTENTS

Page

CHAPTER Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

1 INTRODUCTION

1.1 Backgrounds

Many aspects of engineering practice involve collecting, working with, and using data to solve a problem [1]. The topic of data uncertainty is often called error analysis, a phrase that implies that some error has been made in the measurement or calculation [2]. Uncertainties originate from the limited availability or accuracy of data, from the approximations in modeling or the adequacy of mathematical predictions, from the limited precision of computational processing, and so on. The systematic uncertainty quantification affecting physical systems is critical for engineering design and analysis, where risks must be reduced as much as possible [3]. Determining appropriate ways of dealing with uncertainties has been a constant challenge in engineering. Statistical methods provide a powerful aid to describe and understand uncertainties. Statistics is the branch of applied mathematics concerned with variability and its impact on decision-making [3]. This work focuses on applying Bayesian methods and machine learning in uncertainty quantification and prognostics among all the statistical methods.

Bayesian data analysis is a practical method for making inferences from data using probability models for quantities being observed and for quantities to be learned. The essential characteristic of Bayesian methods is their explicit use of probability for quantifying uncertainty in inferences based on statistical data analysis. The process of Bayesian data analysis can be idealized by dividing it into the following three steps: Frist. Set up a full probability model - a joint probability distribution for all observable and unobservable quantities in a problem. Second, conditioning on observed data, calculate and

interpreting the appropriate posterior distribution - the conditional probability distribution of the unobserved quantities of interest, given the observed data. Third, evaluate the fit of the model and the implications of the resulting posterior distribution. A benefit of using the Bayesian method is that it facilitates a common-sense interpretation of statistical conclusions. For instance, a Bayesian (credible) interval can be regarded as having a high probability of containing the unknown quantity, in contrast to a frequentist (confidence) interval, which may strictly be interpreted only in relation to a sequence of similar inferences that might be made in repeated practice [4].

Besides Bayesian statistics, a large portion of this study focuses on applying and developing machine learning models for solving engineering problems. Today, artificial intelligence (AI) is a thriving field with many practical applications and active research topics [5]. Machine learning is an important application of AI that allows machines to learn from data. The learning problems can be roughly categorized as supervised, unsupervised, and reinforcement learning. In supervised learning, the goal is to predict the value of an outcome measure based on several input measures. There is no outcome measure in unsupervised learning, and the goal is to describe the associations and patterns among a set of input measures [6]. Reinforcement learning is useful for learning how to act or behave when given an occasional reward or punishment signals [7]. Great success has been achieved in solving engineering problems using machine learning methods. However, when it is applied to some specific problems, issues are still unsolved, for example, sparse data, missing data, and bad extrapolation behavior. In this study, those issues are explored with engineering applications.

This study focuses on the mechanical properties of materials, both static and fatigue. For static properties, the objective is to predict the strength and quantify its uncertainty for aging materials by surface information fusion. For fatigue properties, this work presents a framework for fatigue life prediction and uncertainty quantification for different types of fatigue loadings. While applying statistical methods for solving the above problems, the existing models are extensively explored, modified, and improved. New models are proposed for specific issues existing in the focused field.

1.2 Objectives and Contributions

Based on the discussion above, the research objectives and contributions are summarized below:

1. The mechanical properties of pipelines installed decades ago decline with time. Maintaining the safety of vintage pipelines requires the accurate estimation of the strength. The objective is to predict the reliability-based strength using nondestructive multimodality surface information. Bayesian model averaging (BMA) is implemented for fusing multimodality non-destructive testing results for gas pipeline strength estimation. Several incremental improvements are proposed in the algorithm implementation.
2. Sparse data observation is very common in fatigue testing due to various reasons, such as time and budget constraints, and availability of testing materials and resources. The objective is to develop a statistical uncertainty quantification method for fatigue S-*N* curves with sparse data. Hierarchical Bayesian data augmentation (HBDA) is proposed to integrate hierarchical

Bayesian modeling (HBM) and Bayesian data augmentation (BDA) to deal with sparse data problems for fatigue *S*-*N* curves.

3. The third objective is to develop a physics-guided machine learning model to overcome limitations in parametric regression models and classical machine learning models for fatigue data analysis. A Probabilistic Physics-guided Neural Network (PPgNN) is proposed for probabilistic fatigue *S*-*N* curve estimation. This model is further developed for fatigue data analysis of additively manufactured Ti-6Al-4V with missing data. Next, a Physics-guided Mixture Density Network (PgMDN) is proposed, which relaxes the limitation of restricted neural network architecture and distribution forms.
4. Multi-fidelity data exist in almost every engineering and science discipline, which can be from simulation, experiments, and a hybrid form. The objective is to combine two or multiple sources of different fidelity data to have a high accuracy summary and low computational cost. Multi-fidelity Data Aggregation using Convolutional Neural Networks (MDA-CNN) is proposed for multi-fidelity modeling. The proposed framework treats the multi-fidelity data as image data and processes them using CNN, which is very scalable to high dimensional data with more than two fidelities.

1.3 Organization of the Dissertation

The dissertation is organized as follows:

Chapter 2 concentrates on uncertainty quantification and prediction based on Bayesian methods. Objectives No. 1 and 2 in Section 1.2 are included in this chapter.

Experimental data collected from the industry are first introduced for probabilistic bulk strength prediction from surface measurements. Candidate models are generated based on a random combination of linear variables. Next, the Bayesian model averaging approach is introduced. Several difficulties of implementing BMA in practice are illustrated in the context of strength estimation. Following that, the overall method is demonstrated using the models and data for vintage pipeline materials. The usefulness of different surface information is analyzed. The advantages of the BMA are discussed based on the results. For probabilistic fatigue stress-life curves prediction with sparse data, first, the strategies for probabilistic *S*-*N* curve estimation with sparse data are developed based on the hierarchical Bayesian model and Bayesian data augmentation. The theories of HBM and BDA are illustrated, and their implementation is specified. Next, the proposed four strategies are validated using fatigue test data from the literature. Following that, an engineering application is given to estimate the probabilistic *S*-*N* curves of the demolished Pearl Harbor Memorial Bridge, where only limited samples are available for testing. Several issues related to this research are discussed.

Chapter 3 concentrates on uncertainty quantification and prediction based on physics-guided learning. Objective No. 3 is included in this chapter. First, a brief introduction to neural networks is given with basic terminologies and math formulations. Next, a probabilistic method for *S*-*N* curve estimation using the neural network is proposed. The physics-guided machine learning method is described, including physics-based constraints on the neural network parameters and the construction of the neural network architecture. Following that, extensive experiments are conducted for model validations

considering both single factor and multi-factor. Detailed discussions are given for the effect of encoded physics constraints on the results. The necessities of the physics guidance in the neural network for the fatigue data analysis are illustrated with examples where issues occur if the classic neural network is employed. Next, the probabilistic physic-guided neural network (PPgNN) is applied in the fatigue data analysis of additively manufactured Ti-6Al-4V. The PPgNN is adapted to be capable of learning from data set with missing data. Following the methodology, the database collected from open references is described. Next, the results are shown according to predictive performance and improvement with incomplete data. Following this, the Physics-guided Mixture Density Network (PgMDN) is proposed. A brief introduction of Mixture Density Networks is presented. Next, the method for incorporating physics constraints in Neural Networks is introduced. An algorithm for NN training (i.e., constrained optimization) is described. Next, the model is illustrated using a numerical example and a practical problem involving fatigue survival analysis. Some discussions are presented.

Chapter 4 proposes a framework for prognostics based on multi-fidelity learning. Objective No. 4 is included in this chapter. First, the methodology of Multi-fidelity Data Aggregation using Convolutional Neural Networks (MDA-CNN) is presented. Next, the MDA-CNN is validated with extensive numerical examples. Following that, the MDA-CNN is applied in two engineering problems, stress prediction with finite element analysis and fatigue crack growth. After that, Discussions are given to illustrate the benefits and limitations of the proposed framework.

2 UNCERTAINTY QUANTIFICATION AND PROGNOSTICS BASED ON BAYESIAN METHODS

2.1 Overview

In this Chapter, Bayesian methods are applied in uncertainty quantification and prognostics in engineering. Section 2.2 focuses on yield strength prediction, and Section 2.3 describes a method for probabilistic fatigue stress-life estimation.

Section 2.2 focuses on the reliability-based strength prediction using nondestructive multimodality information by the method of Bayesian model averaging (BMA). A class of models is formed from all cases of linear combinations of the surface property measurements. The models are averaged based on the posterior model probabilities. Occam's window is introduced to reduce the number of models under consideration while keeping the predictive accuracy. By not conditioning on any single model, BMA provides more reliable strength prediction by accounting for model uncertainties. In addition, the usefulness of the variables used to predict the strength is evaluated according to the frequency of appearance in the models with high posterior probabilities. The variables with paramount predictive importance can be selected in this way. Thus, the BMA method shows advantages in both vintage pipe strength prediction and model selection [8].

In this Section 2.3, a novel statistical uncertainty quantification (UQ) method for fatigue *S-N* curves with sparse data is proposed. The method called Hierarchical Bayesian data augmentation (HBDA) is proposed to integrate the hierarchical Bayesian modeling (HBM) and Bayesian data augmentation (BDA) to deal with sparse data problems specifically for fatigue *S-N* curves. The key idea is to use: (1) HBM for analyzing the

variability of *S*-*N* curves both within one stress level and across stress levels; (2) BDA to build up a large-size sample of fatigue life data based on the observed sparse samples. Four strategies to estimate the probabilistic *S*-*N* curves with sparse data are proposed: (1) hierarchical Bayesian modeling (HBM) only, (2) Bayesian data augmentation (BDA) only, (3) posterior information from HBM used as prior information for BDA (HBM+BDA), and (4) augmented data from BDA used by HBM (BDA+HBM). Strategy (3) and (4) are named HBDA hereafter. Next, the four strategies are validated and compared using aluminum alloy data and laminate panel data from open literature. Convergence study and confidence estimation are performed, and it is shown that the HBDA methods (i.e., HBM+BDA or BDA+HBM) have better performance compared with the classical method and HBM/BDA alone. The performance gain is especially significant when the number of available data samples is small. Finally, the proposed methodology is applied to a practical engineering problem for fatigue property quantification of the demolished Pearl Harbor Memorial Bridge, where only limited samples are available for testing [9].

2.2 Probabilistic Bulk Strength Prediction from Surface Measurements Using Bayesian Model Averaging

2.2.1 Introduction

The United States is one of the leading nations with the most mileage length in the gas pipelines, especially those of early decades of construction. The latest data published by Pipeline and Hazardous Materials Safety Administration in 2017 shows that the US has recorded a total of 2,541,894 miles of gas pipelines across the nation. Taking natural gas transmission pipelines, for example, 25 percent of the nation's total energy consumption is

transported by this way and delivered to more than 70 million U.S. consumers. It is vital to ensure the safety and reliability of the pipelines. For the pipelines installed decades ago, some of the properties are not known with adequate certainty due to incomplete record-keeping, mergers, and acquisitions, and/or less rigorous testing prior to the enactment of the federal safety codes. Yield strength is one such property that also is critical for design basis, fitness-for-service, leak vs. rupture criteria, and maximum allowable operating pressure determinations. Yield strength forms the basis of critical stress calculations along with the loads and pressures on the pipelines. However, it is a common problem to estimate the mechanical properties due to the lack of available data sometimes. The operators may be forced to either assume a very conservative default value or need to re-establish the material properties through pipeline shutdown, cutout of samples, and then lab testing.

Accurate pipe material strength estimation is critical for the integrity and risk assessment of the pipeline infrastructure systems [10, 11]. In order to measure the mechanical properties of the pipelines without shutdown and cutouts and repairs, a reliable non-destructive set of test methods would be of great benefit. Some indirect methods are proposed through the relationship between the yield strength and surface material properties such as chemical composition, volume fraction of microstructures, and hardness [12]. Several models are proposed based on different material measurements [13-16]. Each type of measurement contains a certain amount of valuable information for pipe strength estimation. Also, they are all associated with different amount of uncertainties. Relying on a single model often leads to predictions that have good performance in some situations at the expense of others. Therefore, there is a strong need to fuse multimodal information in

a statistically meaningful way for more accurate estimation and a robust modeling approach that will work for many types of vintage pipeline steels and their chemistries, microstructures, percent cold work, and physical properties like grain size.

The general method for an ensemble of multiple models is to linearly combine individual model predictions according to different weighting strategies [17]. The weights can be equal for all models in the simplest case. Also, they can be determined through certain regression-based methods. In this way, the weights obtained are actually regression coefficients and hard to interpret since they take on arbitrary positive or negative values and cannot represent the performance of each model [18]. Another method to calculate the weights of models is based on information criterion [19], for example, Akaike information criterion (AIC, AICc) and Bayesian information criterion (BIC). However, different information criteria may generate different preference ranks among models since the best model selected by either AIC or BIC can be distinctly different [20]. Attention should be carefully paid to the choice of information criterion.

Recently, Bayesian Model Averaging (BMA) has gained popularity in diverse fields, such as management science, medicine, and meteorology [17, 21]. BMA is an extension of the usual Bayesian inference methods in which one models both parameter uncertainty and model uncertainty through obtaining posterior parameter and model posteriors using Bayes' theorem. Thus, BMA allows for direct model selection, combined estimation, and prediction [22]. The BMA weights, all positive and summing up to one, reflect relative model performance since they are the probabilistic likelihood measure of a model being correct given the observations.

This study explores the use of BMA for strength predictions by fusing surface information of the vintage pipeline materials. The models being considered are constructed by randomly combining the multimodality surface measurements from the nondestructive testing, with each case of linear combinations being included. The weight assigned to each single model is achieved by Bayesian model regression and model posterior probability calculation. An ensemble model is obtained by averaging the candidate models. To avoid an extremely large number of candidate models, Occam's Window algorithm [23] is applied to average a subset of models that are supported by the data. The predicted averaged strength distributions are compared with those of single models and the performance of BMA is evaluated.

2.2.2 Data and Models

2.2.2.1 Data

Data are collected from 20 existing vintage gas pipes. The data of each pipe fall into 5 categories (Fig. 2.1): yield strength, chemical composition, grain size, hardness, and volume fraction. The yield strengths are measured from both surface indentation technique and full-wall (pipe wall thickness) tensile testing. The strengths from the surface measurements are close to the experimental strengths but with some deviations. The chemical composition is measured in weight percentage using a portable spectrometer after 0.003 to 0.005 inches of surface grinding. Elements include carbon (C), manganese (Mn), phosphorus (P), sulfur (S), aluminum (Al), chromium (Cr), copper (Cu), molybdenum (Mo), niobium (Nb), nickel (Ni), silicon (Si), titanium (Ti), and vanadium (V). The grain sizes are measured at about 0.005 inches from the surface and in mm. The Knoop hardness

values are taken at 0.005 inches from the surface. The volume fraction is represented by percentage pearlite estimated by the lever rule using the carbon content at the surface.

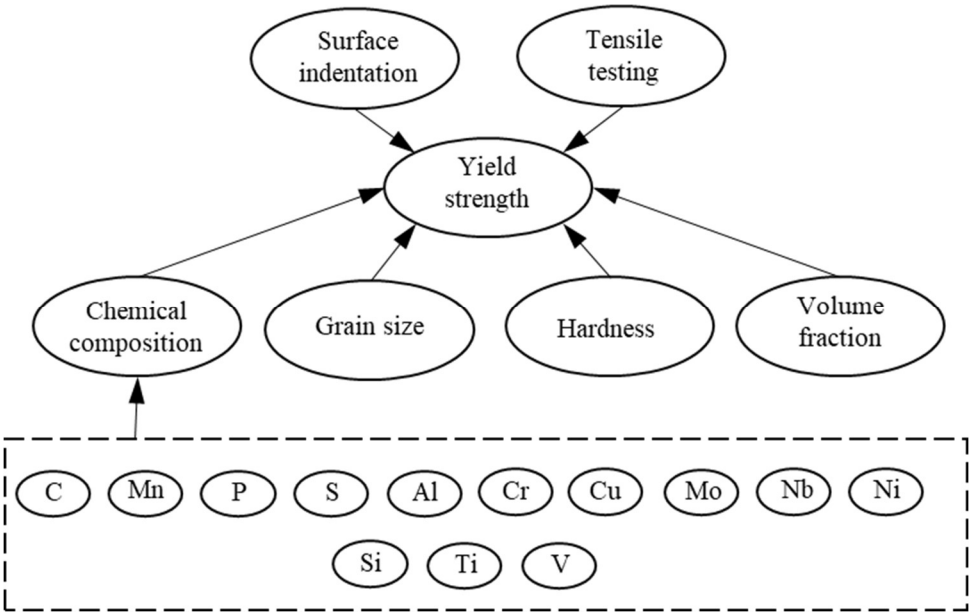


Fig. 2.1 Data Categories.

2.2.2.2 Models

In Ref. [14], the yield strength of steel is expressed as the linear combination of chemical composition and grain size to the power -1/2. [13] predicts the yield strengths of two-phase steel by a linear model of volume fraction. Similarly, a linear prediction model of yield strength with hardness is proposed in [15]. Based on these existing studies, a general model is proposed as

$$YS_T - YS_S = f(CC, GS^{-1/2}, HD, VF)\,. \tag{2.1}$$

where YS_T and YS_S are yield strength measured by tensile testing and surface indentation technique, respectively, CC is chemical composition, GS is grain size, HD is hardness, and VF is the volume fraction of pearlite. The function $f(.)$ is a linear model. A total of 16

variables exist in Eq. (2.1). The yield strength measured by the surface indentation YS_S serves as the baseline and then results of other non-destructive measurements (i.e., chemical composition, grain size, etc.) are used to correct/refine YS_S to obtain a more accurate estimate of YS_T. This is not the only feasible way for information fusion. The authors choose this way as the surface indentation already provides a relatively accurate bulk strength estimation and is currently being used as a nondestructive testing method for strength estimation. Thus, any improvement from other sources of information can be easily justified when the surface indentation prediction is considered as the baseline.

Eq. (2.1) is a full model which contains all the variables but may not be the best model supported by the data. This is because simpler explanations are to be preferred unless there is sufficient evidence in favor of more complicated explanations [24]. Therefore, a class of models to be considered is the collection of all the counterparts of Eq. (2.1). These models are the reduced models of the full model together with the Eq. (2.1). In another word, the models considered are linear models of any variable CC (C, Mn, P, S, …), $GS^{-1/2}$, H, and VF or any linear combination of these variables.

2.2.3 Bayesian Model Averaging

Suppose there is a set of "reasonable" models M_1, …, M_k for estimating a quantity μ from the set of data y. Instead of using one single model for reaching inference for μ, Bayesian model averaging constructs the posterior density of μ given data, $\pi(\mu \mid y)$, not conditional on any model. This is arrived at via the Bayes formula [18, 21, 25, 26].

With the prior information on the parameter given the model, the integrated likelihood or marginal likelihood $\lambda_{n,j}(y)$ for model M_j is expressed as

$$\lambda_{n,j}(y) = \int f(y \mid M_j, \theta_j)\pi(\theta_j \mid M_j)d\theta_j \,. \tag{2.2}$$

where $f(y \mid M_j, \theta_j)$ is the likelihood of data given the model M_j and its parameters θ_j, $\pi(\theta_j \mid M_j)$ is the prior densities for parameters θ_j of model M_j. The marginal likelihood is larger for a model if more of its parameter space is likely and smaller for a model with large areas in its parameter space having low likelihood values. Even if the likelihood function has high peaks, to increase the marginal likelihood these peaks must compensate for the areas in its parameter space where the likelihood is low. Thus, a simpler model with compact parameter space will have a larger likelihood than a more complicated model, unless the latter is significantly better at explaining the data [27].

Using the Bayes theorem, the posterior density of the model is obtained as

$$P(M_j \mid y) = \frac{P(M_j)\lambda_{n,j}(y)}{\sum_{j'=1}^{k} P(M_{j'})\lambda_{n,j'}(y)} \,. \tag{2.3}$$

where $P(M_j)$ is prior probabilities for models $M_1, \ldots, M_k$ under consideration. After computing the posterior density $\pi(\mu \mid M_j, y)$ of μ for each model assuming that M_j is true, the averaged posterior distributions weighted by posterior model probability is given by

$$\pi(\mu \mid y) = \sum_{j=1}^{k} P(M_j \mid y)\pi(\mu \mid M_j, y) \,. \tag{2.4}$$

Eq. (2.4) shows that the posterior density $\pi(\mu \mid y)$ is a weighted average and mixture of the conditional posterior densities, where the weights are the posterior probability of each model. By not conditioning on any given model, BMA avoids the mistake of ignoring model uncertainties [28]. From the properties of mixture distributions, the posterior mean and variance of the averaged model are expressed as

$$E(\mu \mid y) = \sum_{j=1}^{k} P(M_j \mid y) E(\mu \mid M_j, y). \tag{2.5}$$

and

$$\begin{aligned} \mathrm{Var}(\mu \mid y) = & \sum_{j=1}^{k} P(M_j \mid y) \mathrm{Var}(\mu \mid M_j, y) + \\ & \sum_{j=1}^{k} P(M_j \mid y) [\mathrm{E}(\mu \mid M_j, y) - \mathrm{E}(\mu \mid y)]^2 \end{aligned} . \tag{2.6}$$

respectively.

BMA can provide the following benefits [17]. First, the BMA predictions receive higher weights from better-performing models since the likelihood of a model is essentially a measure of the agreement between the model predictions and the observations. Second, the BMA variance is an uncertainty measure of the BMA prediction. It contains two components: the within-model variance and the between-model variance, as shown in the first and the second terms of the right side of Eq. (2.6).

Before implementing BMA theory for model averaging, there exist the following difficulties and some points worthy of attention in practice:

The marginal model likelihood $\lambda_{n,j}(y)$ calculated according to Eq. (2.2) may be analytically intractable. In many cases, no closed-form integral is available [21, 23]. Markov Chain Monte Carlo (MCMC) approach provides a way to overcome this problem. By drawing samples from the required distribution, the integrals in Eq. (2.2) can be evaluated.

Another issue for BMA is the implementation of averaging according to Eq. (2.4) in practice. The number of models considered may be extremely large. As a result,

averaging over the whole class of models is too expensive computationally. One approach is to reduce the number of models by leaving out the models which are not evidently supported by the data. This can be accomplished by introducing Occam's window to the process of BMA [24, 25, 29, 30]. In addition, after the class of models is chosen, how to practically implement BMA shown in Eq. (2.4) remained unsolved. In this work, the model averaging is achieved through the idea of mixture distributions [26, 31].

To judge the efficacy of BMA, a metric is needed for assessing and comparing the performance of BMA and the single models. The purpose of this study is to predict the strength of vintage pipes. Thus, the predictive performance can be used as a meaningful metric to assess a modeling strategy. The logarithmic scoring rule measures the predictive ability by comparing the probabilities each model assigns to the events that actually occur [23, 25, 32, 33].

The above issues are discussed in the following parts of this section.

2.2.4 Marginal Likelihood Calculation

The topic of model uncertainty has begun to come into focus in the recent two decades because of the availability of MCMC computing methods [23]. The integrals in Eq. (2.24) can in general be hard to compute. The MCMC method provides a stochastic method of obtaining samples, which simulates a Markov chain converging to the posterior distributions [34, 35]. After some interaction steps, the chain converges, and its state can then be used as a sample from the desired distribution. Among many available MCMC algorithms, the Metropolis-Hastings algorithm is one of the popular methods [21]. For a basic introduction to the Metropolis-Hastings algorithm, see [36].

Once the samples of model parameters are drawn from their posterior distributions, the marginal likelihood can then be calculated. The following equation [37] is adopted in this work

$$\lambda_{n,j}(y) = \frac{\delta N/(1-\delta) + \sum_{i=1}^{N} f(y \mid M_j, \theta_{ji}) \Big/ [\delta\lambda_{n,j}(y) + (1-\delta) f(y \mid M_j, \theta_{ji})]}{\delta N/(1-\delta)\lambda_{n,j}(y) + \sum_{i=1}^{N} 1 \Big/ [\delta\lambda_{n,j}(y) + (1-\delta) f(y \mid M_j, \theta_{ji})]}. \tag{2.7}$$

where θ_{ji} (i = 1, …, N) are randomly sampled from the posterior density π ($\theta_j \mid M_j, y$) via MCMC sampling, N is the total number of samples θ_{ji} drawn from the posterior distribution for model M_j, δ is a small number (0.01 for example). Since θ_{ji} gives a small likelihood value, the value of summation in the denominator may be quite large and causes an overflow in computation. The Eq. (2.7) is modified by dividing both numerator and denominator by the number of samples N, i.e.

$$\lambda_{n,j}(y) = \frac{\delta/(1-\delta) + \frac{1}{N}\sum_{i=1}^{N} f(y \mid M_j, \theta_{ji}) \Big/ [\delta\lambda_{n,j}(y) + (1-\delta) f(y \mid M_j, \theta_{ji})]}{\delta/(1-\delta)\lambda_{n,j}(y) + \frac{1}{N}\sum_{i=1}^{N} 1 \Big/ [\delta\lambda_{n,j}(y) + (1-\delta) f(y \mid M_j, \theta_{ji})]}. \tag{2.8}$$

to change the summation to mean value. In this way, the incremental average can be adopted to avoid numerical overflow. The estimator $\lambda_{n,j}(y)$ may be evaluated by using a simple and obvious iterative scheme and converges fast.

2.2.5 BMA Implementation

2.2.5.1 Occam's Window

Occam's window provides a method to reduce the number of candidate models for BMA to make the computation more practical and efficient. There are two basic principles in Occam's window. First, if a model predicts the data far less well than the model which has the best predictions, it can be discredited and should no longer be considered. Thus, among several k models only the models belonging to

$$A' = \left\{ M_j : \frac{\max\{P(M_l \mid y)\}}{P(M_j \mid y)} \le C, \quad l = 1, 2, ..., k \right\}. \tag{2.9}$$

are preserved, where M_j is the model selected to be preserved. C is a constant chosen by the data analyst. Second, for nested models, if the smaller model M_0 receives more support from the data than the larger model M_1, the larger models will be excluded. The essential idea of the second principle is shown in Fig. 2.2. 1/20 and 1 are adopted for O_L and O_R respectively in [29]. This means that if there is evidence for M_0 then M_1 is rejected, but rejecting M_0 requires strong evidence. If the evidence is inconclusive, neither model is rejected. [32] shows that adopting 1/20 and 20 for O_L and O_R respectively may improve the predictive performance and thus are also adopted in this work. In this case, $O_L = O_R^{-1,}$ and the second Occam's window principle becomes the same as the first one.

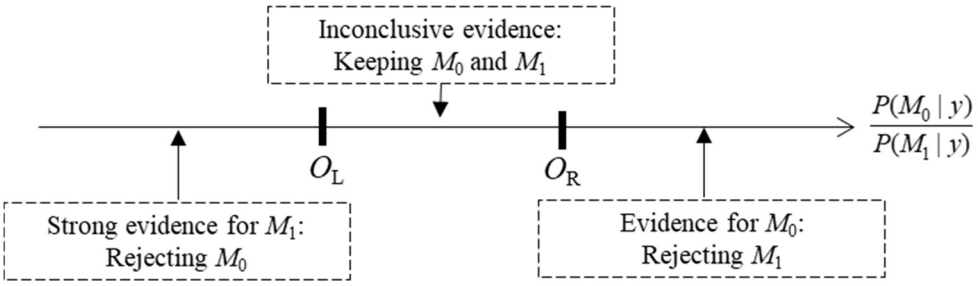


Fig. 2.2 Second Principle of Occam's Window.

2.2.5.2 Mixture Distribution

After the MCMC sampling, a set of parameters θ_{ji} (i = 1, …, N) are sampled from the posterior distributions of parameters of model M_j. Here N is the number of random samples from the posterior distribution of each parameter. Substituting each θ_{ji} into the model, the distribution $\pi(\mu \mid M_j, y)$ of the quantity of interest μ is obtained with the known values of variables. Next, Bayesian model averaging is performed through Eq. (2.4). The posterior density $\pi(\mu \mid y)$ for the averaged model is the mixture distribution of the single conditional posterior density $\pi(\mu \mid M_j, y)$. The mixture distribution based on the weight of each single model is achieved by the following steps.

1. The distributions $\pi(\mu \mid M_j, y)$ form a matrix A with a dimension of $N \times k$ where k is the number of models considered.
2. Taking a sample of the size N, $b = (b_1, \ldots, b_N)$, from the elements of the vector (1, 2, …, k) randomly according to the probability weight. k is the number of models considered. This series of sample b serves as the indexes for mixture distribution.
3. The mixture distribution is obtained by drawing samples from matrix A according to the indexes from Step 2. The ith sample of $\pi(\mu \mid y)$ is extracted from the ith sample of $\pi(\mu \mid M_{bi}, y)$.

An example of the above algorithm is given by a mixture of two normal distributions, N_1 (0, 8) and N_2 (20, 5). The weights for distributions N_1 and N_2 are 0.7 and 0.3 respectively. Fig. 2.3 shows the two single distributions and the mixture distribution.

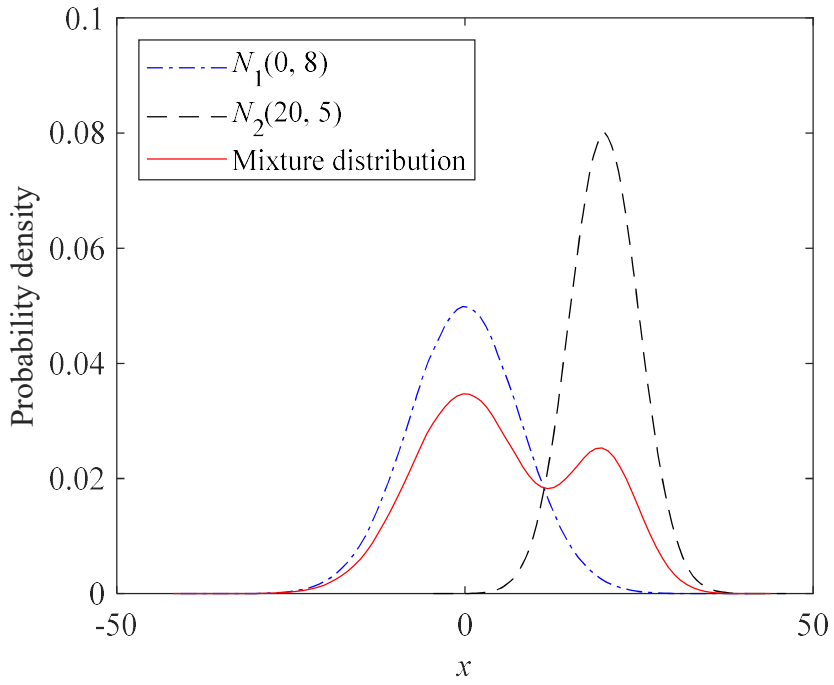


Fig. 2.3 Mixture of Two Normal Distributions.

2.2.5.3 Predictive Performance

A primary purpose of statistical analysis is to make forecasts [18]. Thus, measuring how well a model predicts future observations is one way to judge the efficacy of the BMA strategy. For probabilistic predictions, there exist two types of discrepancies between observed and predicted values: the predictive bias and the lack of calibration [12]. The predictive bias shows a systematic tendency to predict on the low side or the high side. The lack of calibration shows a systematic tendency to over- or understate predictive accuracy. One measure of predictive ability is the logarithmic scoring rule [19] which is a combined

measure of bias and calibration. The smaller the predictive log score for a given model or model average, the better the predictive performance is.

The data are randomly split into two sets. One set of data is called training data (y^B) which is used for Bayesian model regression and averaging. The other set of data called testing data y^T is used to measure the model performance. The predictive log score is related to the expectation of posterior predictive distribution and expressed as [25]

$$-\sum_{d\in y^T}\log E[\pi(d\mid M_j, y^B)]. \tag{2.10}$$

for each single model, and

$$-\sum_{d\in y^T}\log E[\pi(d\mid y^B)]. \tag{2.11}$$

for BMA, where d is one of the testing data, $\pi(d \mid M_j, y^B)$ and $\pi(d \mid y^B)$ is the posterior predictive distribution for model M_j and BMA, respectively. For a single model, $\pi(d \mid M_j, y^B)$ can be obtained by [9, 38]

$$\pi(d\mid M_j, y^B)=\int f(d\mid M_j,\theta_j)\pi(\theta_j\mid M_j, y^B)d\theta_j. \tag{2.12}$$

where $f(d \mid M_j, \theta_j)$ is the likelihood of the testing data, and $\pi(\theta_j \mid M_j, y^B)$ is the posterior density of θ_j given model M_j. After sampling $\theta_{j1}, \theta_{j2}, \ldots, \theta_{jN}$ from $\pi(\theta_j \mid M_j, y^B)$ through MCMC approximations,

$$E[\pi(d\mid M_j, y^B)]=\frac{1}{N}\sum_{i=1}^{N} f(d\mid M_j,\theta_{ji}). \tag{2.13}$$

Next, the log score is calculated by Eq. (2.10). For BMA, $\pi(d \mid y^B)$ is obtained by applying the method of mixture distribution in Section 2.2.5.2. Following that, the log sore of BMA is calculated from Eq. (2.11).

2.2.6 Results and Discussions

2.2.6.1 Variable Selection

A total of 16 variables exist in the general model (Eq. (2.1)) among which 13 variables belong to the chemical composition category. If all the variables are considered in the model class to be averaged, the number of models will be relatively large and computationally expensive since each model requires one simulation [28]. To overcome this difficulty, the following strategy is adopted. First, several relatively important variables in chemical composition are selected based on their evidence for predictive usefulness. Next, the selected chemical components together with grain size, hardness, and volume fraction are used for information fusion and strength prediction.

A total of 8191 models are linearly constructed from the 13 chemical variables. From the total model set, Occam's window method selects 299 models (out of 8191 models) with the posterior model probabilities $P(M_j|y)$ not less than 1/20 of that of the highest posterior probability model. The results of variable selection from the chemical composition are shown in Table 2.1. The first 10 models ranked by posterior model probability $P(M_j|y)$ are listed as M_1, …, M_{10}. $Pr(A_i \neq 0)$ shown in the bottom two rows is used as a metric for evaluating the usefulness of a variable. $Pr(A_i \neq 0)$ is the probability that the coefficient A_i of a variable is non-zero. This is calculated by summing all the posterior probabilities of models which contain this variable. The variable with a higher probability is considered to be more useful. The subscripts all and Occam show that the model set considered in calculating $Pr(A_i \neq 0)$ is the class of all models and the models selected by Occam's window, respectively. From the bottom two rows of Table 2.1, $Pr_{all}(A_i \neq 0)$ is close

to Pr_{Occam} ($A_i \neq 0$), which means that Occam's window method which considers part of the models has similar results as that from the whole model set. The best model $YS_T - YS_S = f$ (Al, Cr, Cu, Nb, Ni) which contains Al, Cr, Cu, Nb and Ni has only 0.67% of the total posterior. This means that there is not a model which is superior to others in the model class and thus substantial uncertainty remains. The interpretation of the Pr ($A_i \neq 0$) was given in [39, 40] and is used in this work. The values between 0.5 and 0.75 are categorized as weak, values between 0.75 and 0.95 as positive, values between 0.95 and 0.99 as strong, and values above 0.99 as decisive evidence. Thus, 0.5 is set as the threshold to have at least weak evidence. For the purpose of variable selection, the probabilities that coefficients are non-zero for P, Cr, Cu, and Si are over 0.5 and higher than the others. As a result, these variables are selected as representatives of the chemical composition for future information fusion with grain size, hardness, and volume fraction.

Table 2.1 Model Selection Only Using Chemical Composition Data.

Model	C	Mn	P	S	Al	Cr	Cu	Mo	Nb	Ni	Si	Ti	V	$P(M_j\|y) \times 10^3$
M1					•	•	•		•	•				6.7
M2						•	•		•	•	•			6.6
M3	•					•	•		•	•				5.6
M4			•			•		•	•		•			5.1
M5			•			•	•		•		•			5.0
M6			•			•	•			•	•			5.0
M7			•	•			•				•		•	4.6
M8			•	•				•	•		•			4.3
M9		•	•	•		•			•		•			4.2
M10			•				•							4.2

Pr_{Occam} ($A_i \neq 0$) × 10	3.1	3.5	5.6	4.5	3.6	5.8	5.6	3.7	4.4	44	5.3	3.6	3.1
Pr_{all} ($A_i \neq 0$) × 10	2.3	2.7	6.1	4.2	2.9	6.2	6.1	3.1	4.2	4.3	5.6	3.0	2.3

2.2.6.2 Results for Full Data

The full data of 20 vintage pipes for the selected chemical composition, grain size, hardness, and volume fraction are used for the Bayesian model selection, averaging, and variable analysis. The results are shown in Table 2.2. A total of 127 models are linearly constructed from the 7 variables (P, Cr, Cu, Si, $GS^{-1/2}$, HD, and VF). From the total model set, Occam's window method selects 37 models (out of 127 models) with the posterior model probabilities $P(M_j|y)$ not less than 1/20 of that of the best model.

Table 2.2 shows the top 10 models ranked by posterior model probability. The best model $YS_T - YS_S = f$(Cu, HD) with the posterior model probability of 0.169 accounts for 16.9% of the total probability, which is more than 3 times higher than that of the second-best model. Another commonly used model selection method is the Bayes information criterion (BIC) [41, 42] based on the maximum likelihood estimation. For the best model, BIC selects the same model as the one with the highest posterior model probability. In addition, it can be observed that the variables of the best model (Cu and HD) have the highest probabilities that their coefficients are non-zero and show positive predictive usefulness. The results provide weak evidence for Cr and indecisive evidence for P, Si, $GS^{-1/2}$, and VF.

Table 2.2 Model Selection Using Chemical Composition Data and Other Features.

Model	P	Cr	Cu	Si	$GS^{-1/2}$	HD	VF	$\lambda_{n,j}(y) \times 10^{23}$	$P(M_j\|y)$	BIC rank
M_1			•			•		13.31	0.169	1
M_2		•	•	•		•	•	4.42	0.056	30
M_3		•	•	•	•	•		4.13	0.053	34
M_4		•	•		•	•	•	4.10	0.052	35
M_5		•		•	•	•	•	3.88	0.049	40

M_6		•	•	•		•		3.45	0.044	11
M_7	•	•		•		•	•	3.27	0.042	50
M_8	•		•					3.00	0.038	3
M_9	•	•	•			•	•	2.81	0.036	54
M_{10}	•		•			•		2.68	0.034	4
	Pr ($A_i \neq 0$)									
BMA	Occam	0.37	0.62	0.77	0.44	0.36	0.78	0.41		
	All	0.34	0.66	0.83	0.42	0.33	0.84	0.39		

2.2.6.3 Predictive Performance - A Specific Case

The results for full data are presented in Section 2.2.6.2 where posterior model probability $P(M_j|y)$ and BIC are applied for model selection using the full data. In this section, the full data are split into training/testing data. The probability $P(M_j|y)$ and BIC are calculated using the training data, and the log scores are calculated using the testing data. This is to evaluate the predictive performance of models selected by $P(M_j|y)$ and BIC using log score. The performance of BMA for predictions of pipe strength is assessed by randomly splitting the data from a total of 20 vintage pipes into two sets: training data (15 pipes) and testing data (5 pipes). The log score serves as the measurement of prediction performance. First, the results of a specific case of data splitting are shown and analyzed in detail. Next, the data are randomly split 100 times. The general predictive performance is evaluated based on these repeated experiments. Table 2.3 shows the results of models selection and averaging for a specific case of data splitting. 33 models are selected in Occam's window out of 127 models. The models listed in Table 2.3 are the top 10 models ranked by posterior model probability. The best model selected by both posterior model probability and BIC is the YS_T - $YS_S = f$ (Cr, Cu). The probability of the coefficients to be non-zero also show weak evidence for Cr and Cu. It should be noted that the model M_2 which performs the best using the full data is ranked as the second (the third by BIC). The model M_2 in Table 2.3 is $YS_T - YS_S = f$ (Cu, HD) which is ranked as the best model in Table 2.2. Table 2.2 used all available data (20 data) to perform the calculation and Table 2.3 only uses partial data (15 out of 20 data) to perform that calculation. Thus, this demonstrates that the ranking depends on the selected datasets. This is because both in the

subset and full set data case, no model shows paramount evidence based on the posterior model probabilities, and thus substantial model uncertainty retains.

Predictive log scores are shown in Table 2.3. BMA shows better predictive performance than the best model M_1. The improvement can be understood in the following way. The log score of BMA is 0.4 less than that of model M_1 for both models in Occam's window and all models. There are 5 data in the testing data set. Therefore the predictive probability of what is actually observed is [exp(0.4/5) -1] × 100% = 8.3% larger for BMA than for best model M_1. However, the model M_2 which is the best model for the full data performs better than the model M_1 and BMA from the predictive point of view.

Table 2.3 Results from the Split Dada.

Model	P	Cr	Cu	Si	$GS^{-}_{1/2}$	HD	VF	$\lambda_{n,j}(y) \times 10^{18}$	$P(M_j\|y)$	Log score	BIC rank
M_1		•	•					5.49	0.189	12.9	1
M_2			•			•		1.89	0.065	11.8	3
M_3		•						1.42	0.049	12.6	2
M_4		•		•	•	•	•	1.35	0.046	12.6	46
M_5		•	•	•		•		1.29	0.044	12.3	16
M_6		•		•				1.26	0.043	12.6	11
M_7		•		•		•	•	1.17	0.040	12.2	29
M_8		•					•	1.10	0.038	13.1	10
M_9		•	•	•	•	•		9.90	0.034	12.5	43
M_{10}	•		•					9.42	0.032	11.8	13
	Pr ($A_i \neq 0$)										
BMA	Occam	0.29	0.73	0.65	0.37	0.27	0.33	0.43			12.5
	All	0.25	0.78	0.69	0.34	0.20	0.27	0.41			12.5

In addition, Fig. 2.4 shows the comparison of predictive performances of direct surface indentation technique, lowest BIC model, highest posterior probability model, and BMA (over all model and the models in Occam's window). The error bars represent the 95% credible intervals. The dots with different shapes are the expectations of different models. To better compare the predictive performances, the root mean squared errors are calculated in Table 2.4. On average, the error of each model is less than that of the surface indentation technique. BMA over all models and the models in Occam's window shows better performance than the lowest BIC model and highest posterior probability model. The expectations of BMA over all models are the best for prediction.

To better compare and analyze the credible interval shown in Fig. 2.4, the predictive distribution of each model is illustrated by taking the data with the largest strength in Fig. 2.4 as an example and shown in Fig. 2.5. It can be seen that the predictive distribution of BMA is wider than that of the single best model selected by BIC or posterior model probability. This is because that conditioning on a single selected model ignores model uncertainty. The best model in Table 2.3 accounts for only 18.9% of the total posterior probability, which shows that model uncertainty is substantial. Ignoring model uncertainty can lead to the underestimation of standard deviation which makes decisions riskier [23, 32]. It should be noted that the values in Fig. 2.4 and Fig. 2.5 are scaled by a constant value to mask the true values.

The posterior model probability $P(M_j|y)$ is the criteria for model selection with all provided data. The calculation of $P(M_j|y)$ is a little complex and the BIC gives an asymptotic approximation to the posterior model probability when the sample size is large enough. If the BIC approximation is sufficient, the model with the best BIC score is expected to be the same as the model that has the highest posterior probability [43]. They are mainly used in Section 2.2.6.2 to identify the highly ranked models. It should be noted that all available experimental data are used in these two criterion calculations as the main purpose is “Which models best explain the observed data?".

The log score is mainly used to check the model’s predictive performance. All available experimental data are split into training and testing data. The testing data is used to calculate the log score for model predictive performance checking. The log score is mainly used in section 4.3 for model predictive performance checking. For example, Tables 6-7 are evaluated using the log score. It should be noted that the model with the highest log score may be different from the model with the highest posterior probability/BIC as different sample size is used (i.e., testing data only vs. all available data).

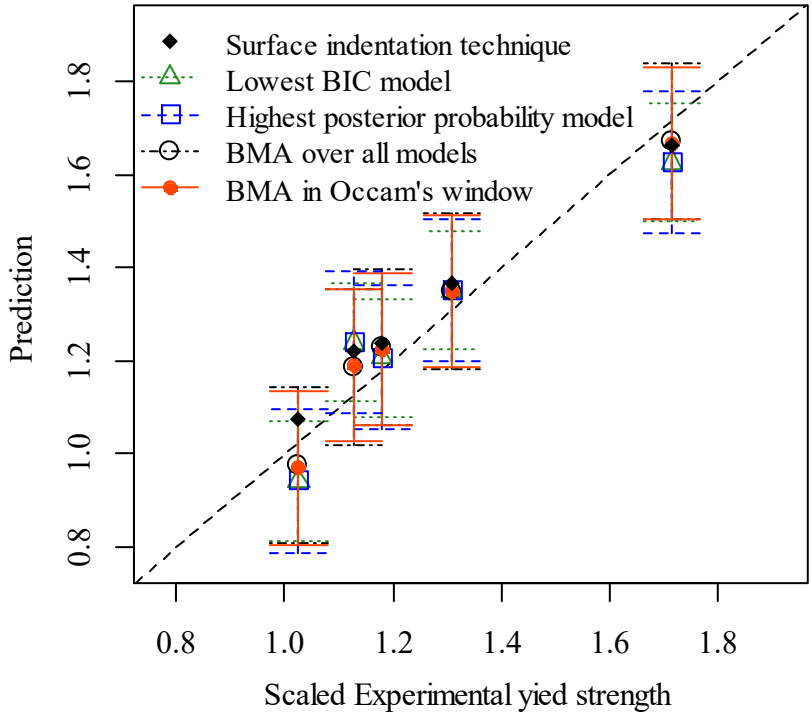


Fig. 2.4 Predictive Performance Comparison.

Table 2.4 Predictive Performance Comparison by Root Mean Squared Error.

	Root mean squared error
Surface indentation technique	5.80
Lowest BIC model	6.62
Highest posterior probability model (expectation)	6.90
BMA over all models (expectation)	4.36
BMA in Occam's window (expectation)	4.61

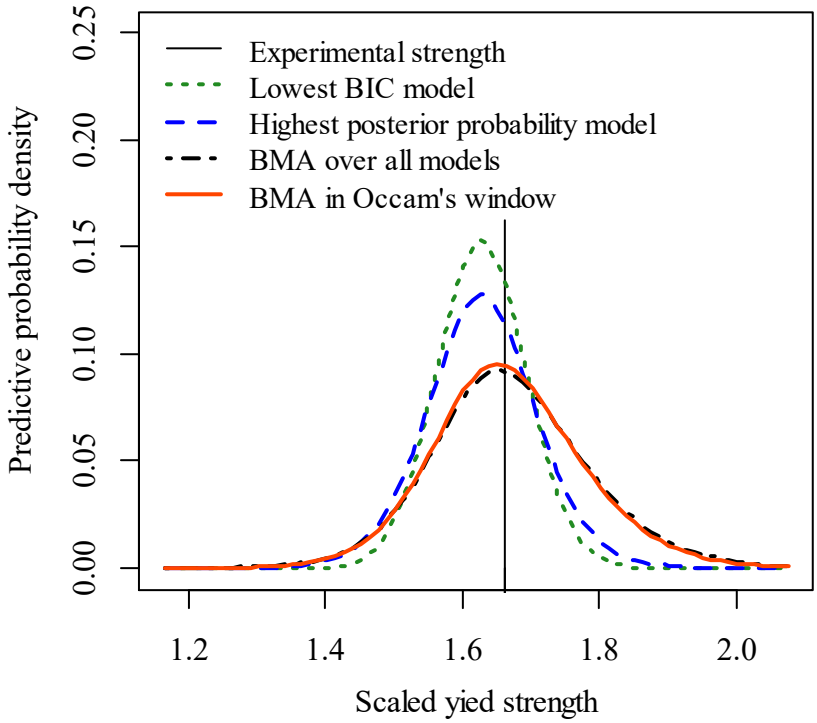


Fig. 2.5 Predictive Distributions.

2.2.6.4 Predictive Performance - General Predictive Performance

In order to study the predictive performance in a more general way, the analyses are repeated for 100 random training/testing data splits to check that the results in Section 2.2.6.3 are not due to the particular random training/testing split used. Table 2.5 shows the average predictive usefulness of variables. Cu and HD with probabilities of coefficients being non-zero are 0.66 and 0.65, respectively (0.65 and 0.63 for BMA over all models), which again shows more positive evidence for prediction than other variables. Although Cr and Cu are the variables preferred in the single case stated above, the results averaging over the 100 times of validations agree with that of the full data. However, for the full data,

the probabilities of coefficients being non-zero are 0.77 and 0.78, respectively (0.83 and 0.74 for BMA over all models) shown in the bottom two rows of Table 2, which are higher than the averaged results. This is due to the fact that the results shown in Table 2.5 are based on 15 training data which are less than the full data. The full data contain 20 data points, among which 15 are used for training and 5 for testing. Due to the training/testing data split, only 15 training data are used for variable usefulness evaluation in Table 2.5. As a result, there is more model uncertainty in a single prediction experiment. By averaging the results of repeated experiments, the effect of this uncertainty can be reduced.

Table 2.5 Average Predictive Usefulness of Variables Over 100 Times of Validations.

		P	Cr	Cu	Si	$GS^{-1/2}$	HD	VF
Pr ($A_i \neq 0$)	Occam	0.47	0.54	0.66	0.45	0.43	0.65	0.42
	All	0.48	0.54	0.65	0.47	0.45	0.63	0.43

Table 2.6 shows the predictive comparison based on the log score between BMA and the model with the highest posterior probability. The average predictive log sores are 15.07, 14.44, and 14.26 for the highest posterior probability model, BMA over models in Occam's window and BMA over all models, respectively. On average BMA can predict the strength with an improvement of 13.5% (in Occam's window) and 17.6% (over all models). Among the repeated experiments of 100 times, BMA both over models in Occam's window and over all models have better predictive ability for 71 times and worse

for 29 times than the highest posterior probability model. In order to see more details of the predictive probability of what is observed for yield strength, the predictive performance improvement or decrease is divided into several ranges from less than 10% to larger than 50%.

When BMA performs better than the highest posterior probability model, the predictive ability is improved from 0 to 20% for nearly a half of times. For the rest, the times when BMA improves the predictive performance are relatively evenly for 20~30%, 30~40%, and 40~50%. In addition, there is a possibility for the predictive performance to be improved to over 50%. The results of BMA over the models selected by Occam's window method and that over all models are close to each other. The difference is that BMA over all models tends to have more times when the improvement falls into higher percentages. Thus, the predictive performance can be improved to a higher degree when all models are considered.

From Table 2.6, it can be seen that model averaging also has the possibility of decreasing the predictive ability and performs worse than the highest posterior probability model. However, the percentages of reduction mostly concentrate in the range of less than 20%. To better understand the cases of BMA being worse than the best model, the model with the highest posterior model probability in that case is studied. It is found that for 25 out of 29 times the highest posterior probability model is YS_T - $YS_S = f$ (Cu, HD) which is

the best model selected by the full data. This model is the best model for only 6 out of 71 times when BMA performs better. It is again shown that variables Cu and HD are of predictive importance. When this model is selected as the best model based on the posterior model probability, the predictive performance of this best model tends to be decreased after being averaged with other models.

Table 2.7 shows the rank of model YS_T - $YS_S = f$ (Cu, HD) based on the posterior model probability. Due to the model uncertainty induced from insufficient data, this model may not be selected as the best model. However, it is very likely to be ranked as one of the top models according to Table 2.7. When this model is not selected as the best model, Bayesian model averaging can provide a tool to improve the predictive ability and reliability.

Table 2.6 Predictive Performance Comparison Over 100 Times: BMA vs. Model with Highest Posterior Probability.

Times out of 100		Sum	<10%	10~20%	20~30%	30~40%	40~50%	>50%
Better	Occam	71	23	15	7	10	7	9
	All	71	18	13	8	8	10	14
Worse	Occam	29	14	11	3	0	1	0
	All	29	15	9	3	0	1	0

Table 2.7 Rank of Model YST - YSS = f (Cu, HD) Out of 127 When BMA Performs Better or Worse Than the Highest Posterior Probability Model.

Rank		sum	1	2	3	4	5	6~10	11~20	21~30	31~40	>40
BMA	better	71	6	16	10	5	6	11	13	3	1	0
	worse	29	25	1	1	0	0	1	0	0	1	0

2.3 Probabilistic Fatigue Stress-life Curves Prediction with Sparse Data Using Hierarchical Bayesian Data Augmentation

2.3.1 Introduction

The relationship between the fatigue life of engineering materials and applied stress is an important property for design-for-reliability processes [44-49]. For safe-life design, the laboratory fatigue test data are often presented in the form of the *S*-*N* curve, e.g., a log-log plot of stress *S* versus the median fatigue life *N* which is expressed in cycles to failure [16, 50-53]. Fatigue lives under the same stress levels typically display a huge scatter range even in well-controlled experiment conditions [54]. In order to describe the probabilistic fatigue property, an extension of the median *S*-*N* curve is to use the *p* quantile *S*-*N* curves, also called probabilistic *S*-*N* curves, a generalization that relates the *p*-quantile of fatigue life to the applied stress. Each curve represents a constant probability of failure [44, 55]. A conventional method to quantitatively describe the probabilistic *S*-*N* curves is to do probability fitting using approximately 15 specimens at each of the four or five constant

stress levels [54, 56]. Thus, a relatively large sample size is needed to obtain the probabilistic *S-N* curves experimentally. In addition, when the stress approaches the fatigue limit, fatigue testing is very time-consuming. Thus, the classical method is time-consuming and costly, which may be beyond the allowable time and budget constraints in practice. Sometimes, the available testing materials (e.g., remaining materials from a demolished structure) and resources (e.g., testing for very large components) may be limited, and the available experimental observations are sparse as well. Therefore, it is needed to develop a statistical method by which reasonable fatigue probabilistic *S-N* curves can be obtained using sparse data, and the cost for the experiment can be reduced without loss of validity for the uncertainty quantification results.

Although a variety of statistical techniques have been utilized to fit probabilistic *S-N* curves in the past decades, there remain difficulties to obtain a reasonable probabilistic *S-N* equation with satisfactory accuracy in the case of sparse experimental fatigue life data [54]. To deal with the probabilistic *S-N* fitting with sparse data, Xie et al. [54] propose a backwards statistical inference method. This method is to establish a large size of samples from sparse data by converting the fatigue lives tested at different stress levels into equivalent lives at an arbitrary stress level. The assumption is that the fatigue lives of the same specimen tested at different stress levels are at the same percentile of the corresponding fatigue life distributions. Implicitly, the correlation of fatigue lives of the

same specimen at different stress levels is assumed to be unity [57]. Adopting a similar methodology with Ref. [54], Tan [58] proposes a sample aggregation principle. The relationship between fatigue life standard deviation and stress level is derived in accordance with the *S*-*N* equation instead of being assumed as a linear relationship used in Ref. [54]. Gao et al. [59] propose a method dealing with sparse data based on the assumption that the probabilistic *S*-*N* curves can be transformed into straight lines in a double logarithm coordinate system and intersect at one point at the high-stress regime.

This study explores the utilization of hierarchical Bayesian model (HBM) and Bayesian data augmentation (BDA) for probabilistic *S*-*N* curves fitting with sparse fatigue life data. Hierarchical Bayesian model is introduced to analyze fatigue data in [50, 60] where probabilistic *S*-*N* curves are obtained from predictive distributions. HBM uses all the data at different stress levels for mean and variance inference [61], which provides a tool for analyzing the sparse fatigue life data. However, the validity of the hierarchical Bayesian model in the case of sparse data is not discussed in detail, which is addressed in this work. In this work, the Bayesian data augmentation serves as another methodology by treating the sparse fatigue life data problem as a missing data problem [62-66]. The main idea is to build up a large-size sample of fatigue life data based on the sparse samples, hence making probabilistic *S*-*N* curve fitting much easier in mathematical treatment and more efficient in utilizing the information carried by the individual test data [54]. To the

best of the authors' knowledge, this is the first attempt to employ the hierarchical Bayesian data augmentation in the sparse fatigue life data analysis.

In this research, both the hierarchical Bayesian model and Bayesian data augmentation are utilized for probabilistic *S*-*N* curve estimation with sparse data individually. In addition, hierarchical Bayesian data augmentation (HBDA) is proposed in this work by combining HBM and BDA methods to further improve the estimation capability. Two approaches of the combination are proposed. One is to use the posterior information from HBM as prior information for BDA (HBM+BDA), and the other one is using augmented data from BDA for HBM analysis (BDA+HBM). The performances of the four strategies (i.e., HBM, DBA, HBM+BDA, and BDA+HBM) are compared using literature data and are applied to a practical engineering problem for steel bridges.

2.3.2 Methodology

2.3.2.1 Fatigue Probabilistic *S*-*N* Curve Estimation with Sparse Experimental Data

In this research, both the hierarchical Bayesian model and Bayesian data augmentation are utilized for probabilistic *S*-*N* curve estimation with sparse data.

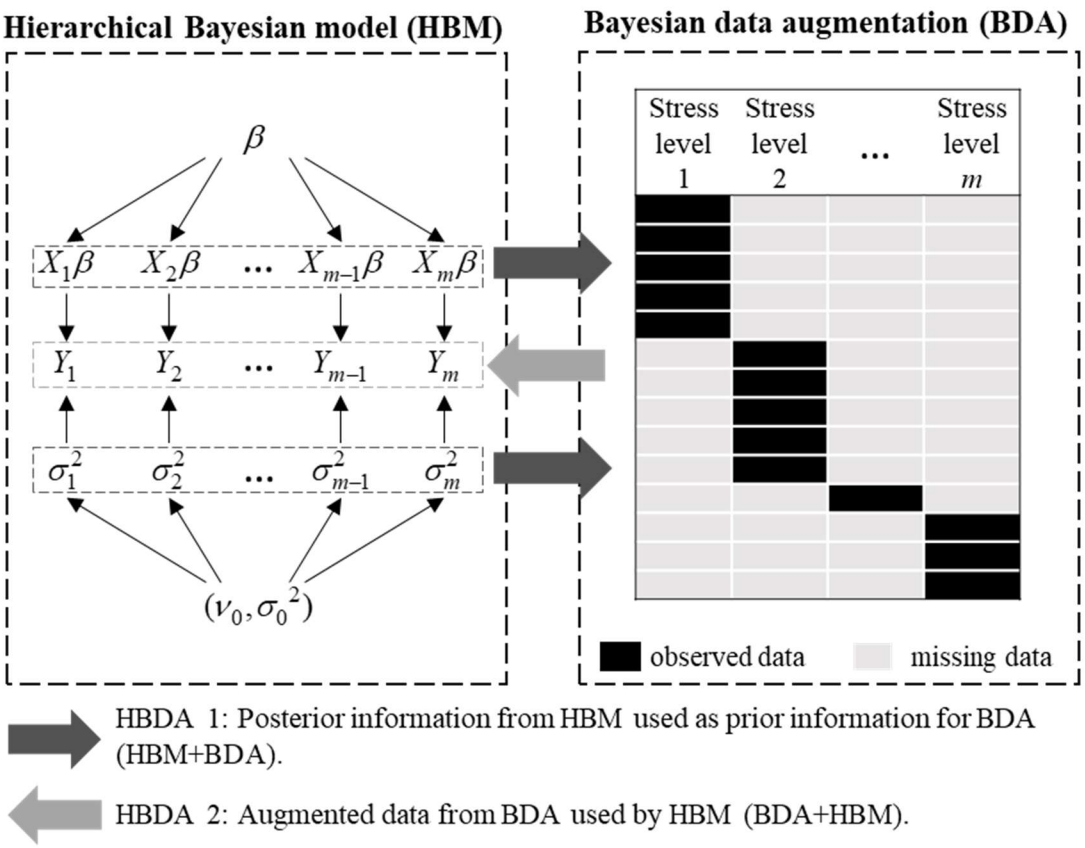


Fig. 2.6 Schematic Overview of the Proposed Method for Fatigue Probabilistic *S-N* Curve Fitting with Sparse Experimental Data.

This work explores the validity of the hierarchical Bayesian model (HBM) in the case of sparse fatigue life data (the left dashed line box in Fig. 2.6). Fatigue life data tested are organized by a hierarchy according to stress levels. The fatigue life variance is generally different at different stress levels. Hierarchical Bayesian modeling can analyze the variability both within one stress level and across stress levels without assuming the monotonic relationships between fatigue life variance and stress level [38, 67]. After hierarchical modeling, the posterior mean and variance for each stress level can be obtained.

Then the predictive distributions in the same hierarchical structure are used to estimate the probabilistic *S*-*N* curves [50].

Another method to generate the probabilistic *S*-*N* is proposed in this work by using Bayesian data augmentation (BDA) based on the missing data theory shown in the right dashed line box of Fig. 2.6. The sparse fatigue life data problem is treated as a missing data problem. That is, a specimen can only be tested to failure at one stress level. Thus, the fatigue lives of this specimen at other stress levels are unknown and are considered as missing data. The theory of missing data [62-66] can then be applied to sparse fatigue data. With data augmented, a large-size dataset can be established and the probabilistic *S*-*N* curves can be obtained by conventional methods used for a large sample.

Also, this work explores the hierarchical Bayesian data augmentation (HBDA), the combination of HBM and BDA, to further improve the accuracy of estimation. The first approach (HBDA 1) to combine these two methods is to use the posterior information from HBM as the prior information for BDA (the right arrows in Fig. 2.6). When using HBM or BDA individually, the prior information is the sample mean and variance. By doing this combination, the sample mean and variance are first updated by considering the across-stress level effect using HBM, and then the sparse data are augmented using BDA based on the updated priors. The probabilistic *S*-*N* curves can be obtained as the method of using BDA individually.

The second approach (HBDA 2) for the HBM and BDA combination is to use the augmented data from BDA for hierarchical modeling (the left arrow in Fig. 2.6). That is, the sparse data are first augmented using BDA. With large sample size data, HBM is implemented to update the sample mean and variance. The probabilistic *S*-*N* curves can be obtained as the method of using HBM individually.

In summary, the following four strategies are investigated in this work (Fig. 2.6):

(1) HBM. The sparse data are analyzed using a hierarchal Bayesian model. The probabilistic *S*-*N* curve is obtained from the predictive distributions.

(2) BDA. A large dataset is established from sparse data using Bayesian data augmentation. The probabilistic *S*-*N* curve is obtained from the augmented data.

(3) HBDA 1: HBM+BDA. First, the sparse data are analyzed using a hierarchical Bayesian model. Next, the posterior mean and variance information from HBM is applied as prior information for Bayesian data augmentation. The probabilistic *S*-*N* curve is obtained from the augmented data.

(4) HBDA 2: BDA+HBM. First, a large dataset is established from sparse data using Bayesian data augmentation. Next, the augmented data are analyzed using the hierarchical model. The probabilistic *S*-*N* curve is obtained from the predictive distributions.

The probabilistic *S-N* curves obtained using each of the above four strategies are compared with that obtained from the conventional method with full data for validation. The detailed theories and implementation of the hierarchical Bayesian model and Bayesian data augmentation are explained in the context of sparse fatigue data in the following of this section.

2.3.2.2 Hierarchical Bayesian Model

The simplest type of multilevel data has two levels, in which one level consists of groups and the other consists of units within groups. For the case of fatigue data, one level is stress level (group), and the other level is fatigue lives of specimens tested at one stress level (units within groups). By denoting $y_{i,j}$ as the data on the ith unit within group Y_j, i.e. $Y_j = \{ y_{1,j}, \ldots, y_{nj,j} \}$, the two levels hierarchical model can be expressed as [38]

$$\{y_{1,j},...,y_{n_j,j} \mid \theta_j\} \sim \text{i.i.d. } p(y \mid \theta_j), \text{ (within-group model)} \tag{2.14}$$

$$\{\theta_1,...,\theta_m \mid \varphi\} \sim \text{i.i.d. } p(\theta \mid \varphi), \text{ (across-group model)} \tag{2.15}$$

$$\varphi \sim p(\varphi), \text{ (prior distribution)} \tag{2.16}$$

where $p(y \mid \theta_j)$ represents variability among measurements within the jth group, $p(\theta \mid \varphi)$ represents variability across groups, $p(\varphi)$ represents information about a single fixed but

unknown quantity. $p(y|\theta_j)$ and $p(\theta|\varphi)$ are sampling distributions estimated from the data, and $p(\varphi)$ is a prior distribution which is not estimated from the data.

From the Basquin relation [68]

$$\begin{aligned} \log(N_j) &= \beta_1 + \beta_2 \log(S_j) + \varepsilon_j \\ \varepsilon_j &\sim \text{i.i.d. normal}(0, \sigma_j^2) \end{aligned}, \tag{2.17}$$

where log denotes the logarithmic function with base 10, the subscript j denotes different stress levels, β_1 and β_2 are fatigue curve coefficients which are shared across stress levels. Rewrite Eq. (2.17) in the matrix form

$$Y_j = X_j \beta + \varepsilon_j, \tag{2.18}$$

where $Y_j = \log(N_j)$, $X_j = (1, \log(S_j))$, and $\beta = (\beta_1, \beta_2)^T$.

In this study, the fatigue lives are assumed to follow a lognormal distribution with different variances at each stress level. Thus, the logarithms of fatigue lives follow a normal distribution. The hierarchical model for describing the heterogeneity of means and variances across populations at different stress levels is the hierarchical normal model, in which the within- and across-group sampling models are both normal [38]. θ_j denotes the parameters at one stress level (i.e. the group-specific mean and variance ($X_j\beta$, σ_j^2)):

$$\{y_{1,j}, ..., y_{n_j,j} \mid X_j\beta, \sigma_j^2\} \sim \text{normal}(X_j\beta, \sigma_j^2) \tag{2.19}$$

(within-group model)

$$\{\sigma_1^2,...,\sigma_m^2\} \sim \text{inverse gamma } (\frac{\nu_0}{2}, \frac{\nu_0 \sigma_0^2}{2}) \quad (2.20)$$

(across-group model)

The full set of unknown quantities includes $\{(X_1\beta, \sigma_1^2), \ldots, (X_m\beta, \sigma_m^2)\}$ representing group-specific means and variances, and (ν_0, σ_0^2) representing the across-group heterogeneity in group-specific variances. The graphical representation of the hierarchical normal model with linear regression can be described by Fig. 2.6 (a).

2.3.2.2.1 Implementation

To obtain the posterior distributions of the unknowns, the Gibbs sampler is implemented by an iterative sampling of each unknown quantity from its full conditional distribution. In the following, the priors and full conditional distributions of the unknowns are given, and the Gibbs sampler implementation is introduced.

The priors used are in the following forms [38]:

$$\beta \sim \text{normal } (\beta_0, \Sigma_0), \quad (2.21)$$

$$p(\nu_0) \propto \exp(-\alpha\nu_0) \quad \nu_0 \sim \text{geometric on the set } \{1,2,...\}, \quad (2.22)$$

$$\sigma_0^2 \sim \text{gamma } (a,b). \quad (2.23)$$

The full conditional distribution is defined as the conditional distribution of one particular parameter given every other parameter's value [69]. A detailed derivation of the full conditional distributions and methods for sampling from them can be found in Refs.

[70, 71]. With the above priors, the full conditional distributions for the unknown parameters are as follows [38]:

$$\begin{aligned} &\{\beta \mid Y_1,...,Y_m, X_1,..., X_m, \sigma_1^2,..., \sigma_m^2\} \sim \text{normal}\,(\beta_n, \Sigma_n) \\ &\Sigma_n = (\Sigma_0^{-1} + mX^T X / \textstyle\sum_{j=1}^{m} \sigma_j^2)^{-1} \\ &\beta_n = \Sigma_n(\Sigma_0^{-1}\beta_0 + mX^T Y / \textstyle\sum_{j=1}^{m} \sigma_j^2) \end{aligned} \quad . \tag{2.24}$$

where $X = [X_1; \ldots; X_m]$, $Y = [Y_1, \ldots, Y_m]$,

$$\begin{aligned} &\{\sigma_j^2 \mid y_{1,j},...,y_{n_j,j}, X_j, \beta, \nu_0, \sigma_0^2\} \sim \\ &\text{inverse gamma}\,(\frac{\nu_0 + n_j}{2}, \frac{\nu_0\sigma_0^2 + \sum_i^{n_j}(y_{i,j} - X_j\beta)^2}{2}) \end{aligned} , \tag{2.25}$$

$$\begin{aligned} &\{\nu_0 \mid \sigma_0^2, \sigma_1^2,..., \sigma_m^2\} \propto \\ &\left[\frac{(\nu_0\sigma_0^2/2)^{\nu_0/2}}{\Gamma(\nu_0/2)}\right]^m \left(\prod_{j=1}^{m}\frac{1}{\sigma_j^2}\right)^{\nu_0/2-1} \exp\left\{-\nu_0\left[\alpha + \frac{1}{2}\sigma_0^2\sum\frac{1}{\sigma_j^2}\right]\right\}, \end{aligned} \tag{2.26}$$

and

$$\{\sigma_0^2 \mid \sigma_1^2,..., \sigma_m^2, \nu_0\} \sim \text{gamma}\,(a + \frac{m\nu_0}{2}, b + \frac{\nu_0\sum_j^m 1/\sigma_j^2}{2}), \tag{2.27}$$

Gibbs sampler is used to draw samples from the full conditional distribution to approximate the posterior distribution of each unknown parameter [38]. Given the current state s of the unknowns $\{\beta^{(s)}, \sigma_1{}^{(s)}, \ldots, \sigma_m{}^{(s)}, \nu_0{}^{(s)}, \sigma_0{}^{(s)}\}$, a new state s+1 may be generated as follows:

Sample $\beta^{(s+1)} \sim p(\beta \mid Y_1,...,Y_m, X_1,..., X_m, \sigma_1^{2(s)},..., \sigma_m^{2(s)})$,

Sample $\sigma_j^{2(s+1)} \sim p(\sigma_j^2 \mid y_{1,j},...,y_{n_j,j}, X_j\beta^{(s+1)}, \nu_0^{(s)}, \sigma_0^{2(s)})$,

Sample $\nu_0^{(s+1)} \sim p(\nu_0 \mid \sigma_0^{2(s)}, \sigma_1^{2(s+1)}, ..., \sigma_m^{2(s+1)})$,

Sample $\sigma_0^{2(s+1)} \sim p(\sigma_0^2 \mid \sigma_1^{2(s+1)}, ..., \sigma_m^{2(s+1)}, \nu_0^{(s+1)})$.

2.3.2.2.2 Predictive Distribution

Bayesian inference makes it possible to obtain the predictive distributions for new observations [38, 72]. The predictive distribution is the conditional distribution of $\tilde{Y}_j$ given the observed data Y_j and can be expressed as

$$p(\tilde{Y}_j = \tilde{y}_j \mid y_{1,j}, ..., y_{n_j,j}) = \int p(\tilde{y}_j \mid \theta_j, y_{1,j}, ..., y_{n_j,j}) p(\theta_j \mid y_{1,j}, ..., y_{n_j,j}) d\theta_j , \qquad (2.28)$$

For probabilistic *S-N* curve estimation purposes, it is useful to have a set of samples of $\tilde{Y}_j$ from its predictive distribution. These samples can be drawn with Gibbs sampler as follows:

Sample $\theta_j^{(1)} \sim p(\theta_j \mid y_{1,j}, ..., y_{n_j,j})$, sample $\tilde{y}_j^{(1)} \sim p(\tilde{y}_j \mid \theta_j^{(1)})$,

Sample $\theta_j^{(2)} \sim p(\theta_j \mid y_{1,j}, ..., y_{n_j,j})$, sample $\tilde{y}_j^{(2)} \sim p(\tilde{y}_j \mid \theta_j^{(2)})$,

…

Sample $\theta_j^{(mc)} \sim p(\theta_j \mid y_{1,j}, ..., y_{n_j,j})$, sample $\tilde{y}_j^{(mc)} \sim p(\tilde{y}_j \mid \theta_j^{(mc)})$.

The sequence $\{\tilde{y}_j^{(1)}, ..., \tilde{y}_j^{(mc)}\}$ constitutes a total of *mc* samples from the marginal posterior distribution of $\tilde{Y}_j$, which is the posterior predictive distribution. The probabilistic *S-N* curves can then be obtained from the predictive distributions [50].

2.3.2.3 Bayesian Data Augmentation

Let Y be the $n \times m$ matrix of all the potential data (complete data), observed and unobserved. Here n and m denote the total number of specimens and stress levels respectively. Let I be the $n \times m$ inclusion matrix, which indexes which potential data are observed. $I_{ij} = 1$ if Y_{ij} is observed and $I_{ij} = 0$ if Y_{ij} is missing. The matrix Y can then be thought of as consisting two parts: $\mathrm{Y}_{\mathrm{obs}} = \{Y_{ij} \mid I_{ij} = 1\}$, the observed data, and $\mathrm{Y}_{\mathrm{miss}} = \{Y_{ij} \mid I_{ij} = 0\}$, the unobserved data.

When considering data collection, it is useful to break the joint probability model into two parts: (1) the model for the underlying complete data, Y, including observed and unobserved components, and (2) the model for the inclusion matrix, I. The complete-data likelihood is defined as the product of the likelihoods of these two factors; that is, the distribution of the complete data, Y, and the inclusion matrix, I, given the parameters in the models [4]:

$$p(\mathbf{Y}, \mathbf{I} \mid \phi, \xi) = p(\mathbf{Y} \mid \phi)\, p(\mathbf{I} \mid \mathbf{Y}, \xi), \tag{2.29}$$

where ϕ and ξ denote the parameter vectors of the distributions of the complete data and the inclusion matrix, respectively. In the context of fatigue data analysis, the estimates of primary interest are functions of the complete data Y and the parameters ϕ. The parameters ξ that index the missingness model are characteristic of data collection but are not generally of interest.

Since Y is not completely observed and the actual information available is (Y_{obs}, I), the appropriate likelihood for Bayesian inference is

$$p(\mathbf{Y}_{\text{obs}}, \mathbf{I} \mid \phi, \xi) = \int p(\mathbf{Y}, \mathbf{I} \mid \phi, \xi) d\mathbf{Y}_{\text{miss}}, \tag{2.30}$$

which is a marginal probability of the observed variables after integrating out the missing variables.

The joint posterior distribution of model parameters ϕ and ξ given the observed information (Y_{obs}, I) is

$$\begin{aligned} p(\phi, \xi \mid \mathbf{Y}_{\text{obs}}, \mathbf{I}) &\propto p(\phi, \xi) p(\mathbf{Y}_{\text{obs}}, \mathbf{I} \mid \phi, \xi) \\ &= p(\phi, \xi) \int p(\mathbf{Y}, \mathbf{I} \mid \phi, \xi) d\mathbf{Y}_{\text{miss}} \\ &= p(\phi, \xi) \int p(\mathbf{Y} \mid \phi) p(\mathbf{I} \mid \mathbf{Y}, \xi) d\mathbf{Y}_{\text{miss}} \end{aligned}, \tag{2.31}$$

The posterior distribution of ϕ alone is this expression averaged over ξ:

$$p(\phi \mid \mathbf{Y}_{\text{obs}}, \mathbf{I}) = p(\phi) \iint p(\xi \mid \phi) p(\mathbf{Y} \mid \phi) p(\mathbf{I} \mid \mathbf{Y}, \xi) d\mathbf{Y}_{\text{miss}} d\xi, \tag{2.32}$$

2.3.2.3.1 Ignorability

When the missing data pattern supplies no information, the data collection mechanism is ignorable. In this case, the posterior distribution of ϕ and the posterior predictive distribution of Y_{miss} are entirely determined by the specification of a data model and the observed values of Y_{obs}. The Eq. (2.32) can be expressed as

$$p(\phi \mid \mathbf{Y}_{\text{obs}}, \mathbf{I}) = p(\phi \mid \mathbf{Y}_{\text{obs}}) = p(\phi) \int p(\mathbf{Y} \mid \phi) d\mathbf{Y}_{\text{miss}}, \tag{2.33}$$

Two conditions are sufficient to ensure the ignorability of the missing data mechanism for Bayesian analysis [4]: (1) The condition of missing at random requires that the distribution of the missing-data mechanism does not depend on the missing values. That is, given ξ, the missingness depends only on Y_{miss}: $p(I \mid Y, \xi) = p(I \mid Y_{miss}, \xi)$. (2) The condition of distinct parameters is satisfied when the parameters of the missing data process are independent of the parameters of the data generating process in the prior distribution: $p(\xi \mid \phi) = p(\xi)$.

In the case of treating the sparse fatigue data as missing data, both of the conditions are satisfied. The reasons are as follows. On one hand, the probability that the fatigue life of a specimen is missing at some stress level A depends only on whether the specimen is tested to failure at other stress level B but not on the value of fatigue life at stress level A. Thus the data are missing at random. On the other hand, the parameters governing the missing-data process are distinct from those of the distribution of the fatigue data, which is typically the case with standard models [4]. Therefore, it is sufficient for obtaining Bayesian inferences without requiring modeling the missing-data mechanism.

2.3.2.3.2 Data Augmentation

The data augmentation process is conducted through Bayesian inference with missing data [73]. Assume the multivariate normal distribution for the logarithms of fatigue life data. For a multivariate normal model with missing data, the unknowns include missing

values as well as model parameters (mean μ and covariance matrix Σ). Gibbs sampling is used to make inferences on these unknowns.

To avoid evaluating the integrals in Eq. (2.33), Gibbs sampling is used to draw posterior simulations of the joint vector of unknown parameters and unobserved quantities, $p(\mu, \Sigma, Y_{miss} \mid Y_{obs})$. Given starting values $\{\Sigma^{(0)}, Y_{miss}^{(0)}\}$, $\{\mu^{(s+1)}, \Sigma^{(s+1)}, Y_{miss}^{(s+1)}\}$ can be generated from $\{\mu^{(s)}, \Sigma^{(s)}, Y_{miss}^{(s)}\}$ by

1. sampling $\mu^{(s+1)}$ from $p(\mu \mid Y_{obs}, Y_{miss}^{(s)}, \Sigma^{(s)})$, the complete-data posterior of μ;

2. sampling $\Sigma^{(s+1)}$ from $p(\Sigma \mid Y_{obs}, Y_{miss}^{(s)}, \mu^{(s+1)})$, the complete-data posterior of Σ;

3. sampling $Y_{miss}^{(s+1)}$ from $p(Y_{miss} \mid Y_{obs}, \mu^{(s+1)}, \Sigma^{(s+1)})$, the conditional predictive distribution of Y_{miss}.

Steps 1 and 2 are together called the posterior or P-step [73, 74]. In steps 1 and 2, the fixed value of Y_{obs} combines with the current value of $Y_{miss}^{(s)}$ to form a current version of a complete data matrix $Y^{(s)}$ having no missing data and a value of the model parameter (μ and Σ) are drawn from $Y^{(s)}$. Step 3 is referred to as the imputation or I-step and corresponds to imputing a value of the missing data Y_{miss} [73, 74]. Step 3 can be expressed in the following:

$$
\begin{aligned}
p(\mathrm{Y}_{\mathrm{miss}} \mid \mathrm{Y}_{\mathrm{obs}}, \boldsymbol{\mu}, \Sigma) &\propto p(\mathrm{Y}_{\mathrm{miss}}, \mathrm{Y}_{\mathrm{obs}} \mid \boldsymbol{\mu}, \Sigma) \\
&= \prod_{i=1}^{n} p(y_{i,\mathrm{miss}}, y_{i,\mathrm{obs}} \mid \boldsymbol{\mu}, \Sigma) \ , \\
&\propto \prod_{i=1}^{n} p(y_{i,\mathrm{miss}} \mid y_{i,\mathrm{obs}}, \boldsymbol{\mu}, \Sigma)
\end{aligned}
\tag{2.34}
$$

For each *i*, the missing elements of the data vector are sampled conditional on the observed elements. This is achieved by the following result about multivariate normal distributions. Let $y \sim$ multivariate normal (μ, Σ), let *a* be a subset of variable indices $\{1, \ldots, p\}$ and let *b* be the complement of *a*. It can be shown from the inverse of partitioned matrices that $\{\mathrm{y}_{[b]} \mid \mathrm{y}_{[a]}, \mu, \Sigma\} \sim$ multivariate normal $(\mu_{b|a}, \Sigma_{b|a})$, where

$$\mu_{b|a} = \mu_{[b]} + \Sigma_{[b,a]}(\Sigma_{[a,a]})^{-1}(\boldsymbol{y}_{[a]} - \mu_{[a]}) , \tag{2.35}$$

$$\Sigma_{b|a} = \Sigma_{[b]} - \Sigma_{[b,a]}(\Sigma_{[a,a]})^{-1}\Sigma_{[a,b]} . \tag{2.36}$$

For a value of *s* that is suitably large, $\mu^{(s)}$, $\Sigma^{(s)}$ and $\mathrm{Y}_{\mathrm{miss}}^{(s)}$ can be regarded as an approximate draw from posterior distributions $p(\mu \mid \mathrm{Y}_{\mathrm{obs}})$, $p(\Sigma \mid \mathrm{Y}_{\mathrm{obs}})$ and posterior predictive distribution $p(\mathrm{Y}_{\mathrm{miss}} \mid \mathrm{Y}_{\mathrm{obs}})$, respectively.

2.3.2.3.3 Multiple Imputation

The key idea of the multiple imputation is to create more than one set of replacements for the missing values in a dataset [66, 75-81]. In multiple imputation, the unknown missing data $\mathrm{Y}_{\mathrm{miss}}$ are replaced by simulated values $\mathrm{Y}_{\mathrm{miss}}^{(1)}$, $\mathrm{Y}_{\mathrm{miss}}^{(2)}$, …, $\mathrm{Y}_{\mathrm{miss}}^{(mi)}$. Each of the *mi* completed datasets is analyzed by standard complete-data methods. The

variability among the results of the *mi* analyses provides a measure of the uncertainty due to missing data [73].

If multiple imputation is to yield valid inferences, the simulated values of μ, Σ and Y_{miss} should be proper Bayesian distributions, that is, the multiple imputations are independent realizations of $p(\mu \mid Y_{obs})$, $p(\Sigma \mid Y_{obs})$, and $p(Y_{miss} \mid Y_{obs})$, respectively. Bayesianly proper multiple imputations reflect both the uncertainty about Y_{miss} given the parameters of the complete-data model and the uncertainty about the unknown model parameters [73].

It is convenient to create multiple imputations using the data augmentation algorithm. Because proper multiple imputations must be independent, the successive iterates of unknowns cannot be used since they tend to be correlated. Rather, the chain needs to be subsampled, e.g. take every *k*th iterate, where *k* is chosen large enough so that the dependence will be negligible [73].

2.3.3 Validation

The proposed method is validated using aluminum alloy 2524-T3 data [54] and laminate panel data [44, 82]. The reference probabilistic *S-N* curves obtained from the conventional method using full data are served as a benchmark. Next, using sparse data, the probabilistic *S-N* curves can be obtained and are compared with the reference one by different strategies, i.e., conventional method, hierarchical Bayesian modeling (HBM),

Bayesian data augmentation (BDA), posterior information from HBM used as prior information for BDA (HBDA 1: HBM+BDA), and augmented data from BDA used by HBM (HBDA 2: BDA+HBM).

The general idea for validation is the following two steps. For both aluminum alloy 2524-T3 data and laminate panel data, the large-sample data (about 15 data at each stress level) are used to obtain the probabilistic *S-N* curves from different strategies and are compared with the reference one. This is to test, in the case of large-sample data, whether the probabilistic *S-N* curves from different methods can yield the same result as the conventional method. Next, small-sample data (5 or 3 data at each stress level) are selected as sparse data. Using the sparse data, the deviations between the probabilistic *S-N* curves using the proposed methods and the reference probabilistic *S-N* curve are checked to test the performance of each strategy. For laminate panel data, the performances of different proposed strategies are studied when the sample size at each stress level changes from 1 to 15. This is to evaluate the validity of each strategy for probabilistic *S-N* curve estimation with the change of data sparseness.

2.3.3.1 Aluminum Alloy Data

About 15 aluminum alloy 2524-T3 specimens are tested at each of 4 stress levels. The fatigue life data from Ref. [54] are shown in Table 2.8. The fatigue lives at each stress level are ordered in the sequence that the randomly selected specimen's life data come into

being [54]. Thus, the first several fatigue life data at each stress level can be chosen from the full dataset to establish a small-sample dataset and are used for validation. 3 scenarios are considered: 14-15-15-15, 5-5-5-5, and 3-3-3-3. The test data corresponding to scenario 14-15-15-15 are all the observations at each of the four stress levels, which are the full data. For scenarios 5-5-5-5-5 and 3-3-3-3-3, the test data are first 5 and 3 observations are each stress level, respectively, which represents the medium data size and sparse data size, respectively.

Table 2.8 Fatigue Life Data of Aluminum Alloy 2524-T3 [54].

S_j	$X_j = \log(S_j)$	n_j	$Y_j = \log(N_j)$
400	2.60	14	4.477, 4.400, 4.426, 4.462, 4.592, 4.411, 4.447, 4.402, 4.665, 4.475, 4.458, 4.551, 4.525, 4.641
350	2.54	15	4.784, 4.842, 4.776, 4.813, 4.813, 4.860, 4.798, 4.776, 4.758, 4.770, 4.755, 4.837, 4.736, 4.842, 4.796
300	2.48	15	5.028, 5.074, 5.016, 4.894, 4.993, 5.071, 5.024, 5.035, 4.954, 5.039, 5.098, 5.057, 5.092, 5.082, 5.005
200	2.30	15	5.603, 5.544, 5.528, 5.630, 5.594, 5.540, 5.581, 5.548, 5.426, 5.567, 5.554, 5.627, 5.630, 5.596, 5.626

The results are plotted in Fig. 2.7. The left three figures of Fig. 2.7 show the probabilistic *S*-*N* curves with the confidence level of 95% and the survivability of 99% for three scenarios 14-15-15-15, 5-5-5-5, and 3-3-3-3 obtained using different strategies: conventional, HBM, BDA, HBM+BDA, and BDA+HBM. The right three figures are the errors at each stress level as well as the root mean square errors (RMSE) of all the stress

levels for different strategies. For errors at each stress level, a positive or negative error value means that the percentile point obtained by some strategy is larger or smaller than that obtained by a conventional method with full data (benchmark). Root mean square errors are all positive.

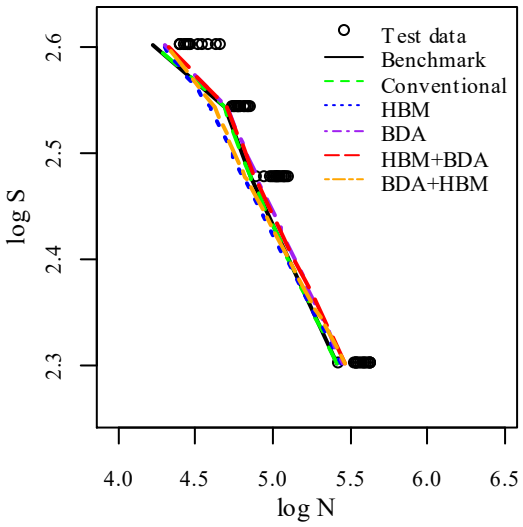
For large-sample data scenario 14-15-15-15 (Fig. 2.7(a) and (b)), the probabilistic *S-N* curves obtained from all the methods nearly coincide with the benchmark curve, the maximum errors at four stress levels are 0, -1.87%, 1.87%, 1.99%, 2.29%, and the RMSE are 0, 1.41%, 1.09%, 1.14%, 1.49% for the conventional method, HBM, BDA, HBM+BDA, and BDA+HBM, respectively. The errors of the conventional method are zeros since all the test data are used for estimation, which produces the benchmark result. This shows that the four strategies proposed in the work for probabilistic *S-N* curve fitting with sparse data are valid in the case of large-sample data.

Fig. 2.7(c) and (d) show the medium-sample data scenario 5-5-5-5. The maximum errors at four stress level are -4.77%, -4.16%, 1.71%, 1.81%, -2.05% and the RMSE are 3.13%, 2.45%,1.27%, 1.10%, 1.72% for conventional method, HBM, BDA, HBM+BDA, and BDA+HBM, respectively. As the number of available data decreases from a large size to a medium one, the overall results produced by the conventional method tend to deviate from the benchmark. However, all four proposed strategies can produce better results than the conventional method. Among all the proposed probabilistic *S-N* estimation strategies,

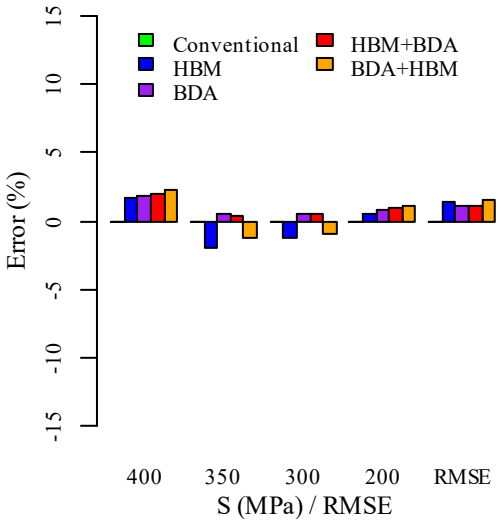
the HBM+BDA method outperforms the others, while the performance of HBM is the worst.

As the dataset shrinks to sparse data (Fig. 2.7(e) and (f)), the maximum errors at four stress level are -8.78%, -6.60%, 3.05%, 1.57%, -2.89% and the RMSE are 7.83%, 4.41%,1.98%, 0.96%, 1.87% for conventional method, HBM, BDA, HBM+BDA, and BDA+HBM, respectively. The probabilistic *S*-*N* curve obtained from the conventional method is too conservative. For sparse data cases, the hierarchical Bayesian model is helpful but will achieve in larger prediction intervals due to more uncertainties caused by the insufficient sample size [2]. Again, the strategy HBM+BDA has the lowest RMSE among all the four strategies, followed by BDA and BDA+HBM. The HBM performs better than the conventional method but is still the worst compared with the other three strategies.

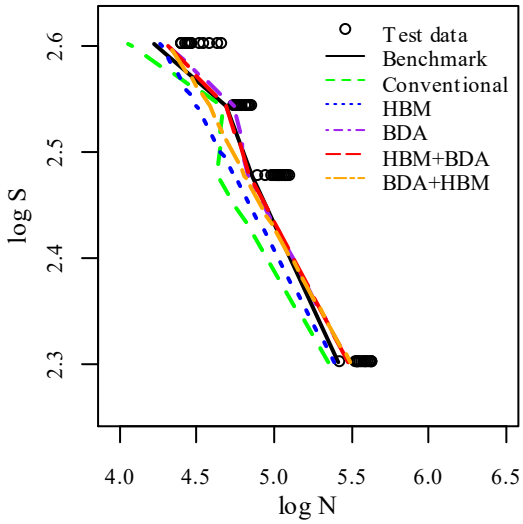
From the results for the aluminum alloy data, it can be concluded that the four strategies proposed in this work HBM, BDA, HBM+BDA, and BDA+HBM can produce better probabilistic *S*-*N* curves compared with the conventional method in the case of sparse fatigue life data. However, the performances with sparse data are different. Among all these four strategies, the HBM tends to produce over-conservative results, due to the uncertainty caused by the small sample size [50]. The HBM+BDA method outperforms the others in this example.



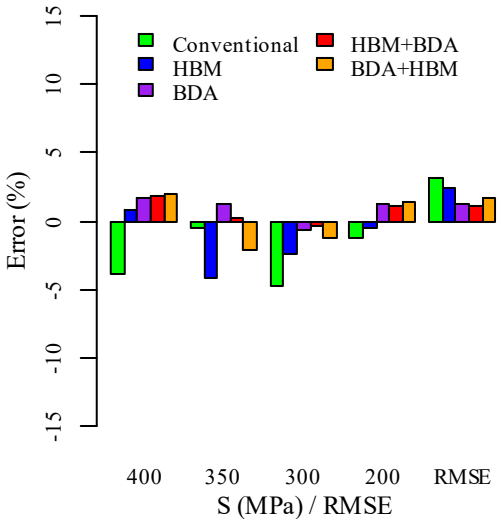
(a) Scenario 14-15-15-15: probabilistic *S-N* curves.



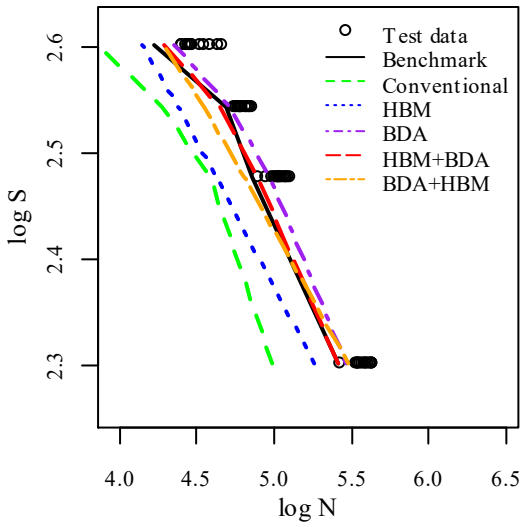
(b) Scenario 14-15-15-15: Errors.



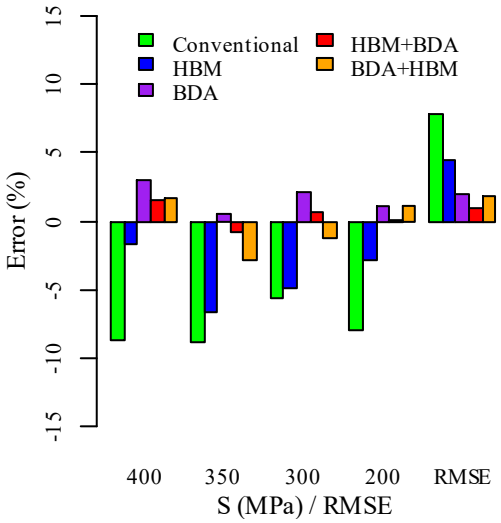
(c) Scenario 5-5-5-5: probabilistic *S-N* curves.



(d) Scenario 5-5-5-5: Errors.



(e) Scenario 3-3-3-3: probabilistic *S-N* curves.



(f) Scenario 3-3-3-3: Errors.

Fig. 2.7 Aluminum Alloy Data: Probabilistic *S-N* Curves with the Confidence Level of 95% and the Survivability of 99% ((a), (c), (e)) and Errors at Each Stress Levels ((b), (d), (f))

2.3.3.2 Laminate Panel Data

In this section, the four strategies dealing with probabilistic *S-N* curve estimation with sparse data are validated using laminate panel data [44, 82]. The specimens are tested at 5 stress levels and the failure data are shown in Table 2.9. The data at each stress level are given in ascending order. Thus, the subset data need to be randomly chosen from the full dataset. The purpose of stating the order of the fatigue life data at each stress level in Table 2.8 and Table 2.9 is to illustrate how to randomly select small-size data set from the large-size data. The data in Table 2.8 is shown in the same order as those from Ref. [54].

Ref. [54] provides the information of the order of the experimental data. The data in Table 2.9 are from Ref. [44, 82] which do not provide the information of the order of experimental data. Thus, the data in Table 2.9 are listed in ascending order. If the order information is given in the literature reference (Table 2.8), the first several experimentally obtained fatigue life data at each stress level can be chosen from the full dataset to establish a small-sample dataset, which is random in nature. Otherwise (Table 2), the small-sample data need to be randomly chosen from the full dataset.

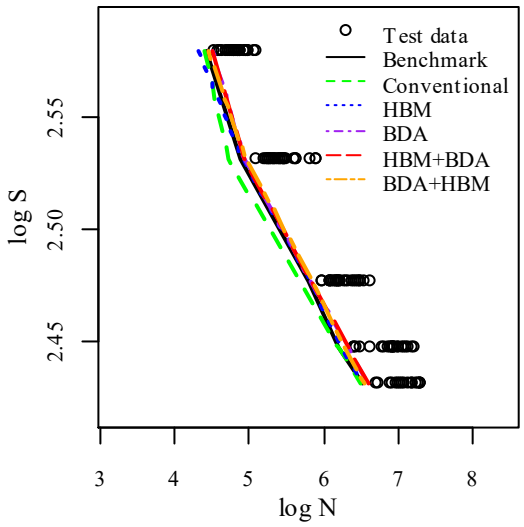
In this validation example, two parts are conducted. First, 3 scenarios are considered as the above example, i.e., large, medium, and sparse data size corresponding to 15-15-15-15-15, 5-5-5-5-5, and 3-3-3-3-3, respectively. The data used for each scenario are randomly picked up. Next, the performance of each strategy as the number of data changes is studied. The scenarios with sample size of 1 to 15 at each stress level are considered, i.e., 1-1-1-1-1, 2-2-2-2-2, …, 15-15-15-15-15. In each scenario, the validation is repeated 50 times, at each of which different subset data are selected.

Shown in Fig. 2.8 are the probabilistic *S-N* curves with the confidence level of 95% and the survivability of 99% for large, medium, and sparse data. Similarly with the above example, in the scenario of 15-15-15-15-15, all the four strategies can produce nearly the same results with the benchmark. As the number of samples at each stress level is reduced, the errors of the results from the conventional method get unacceptably high, while all the

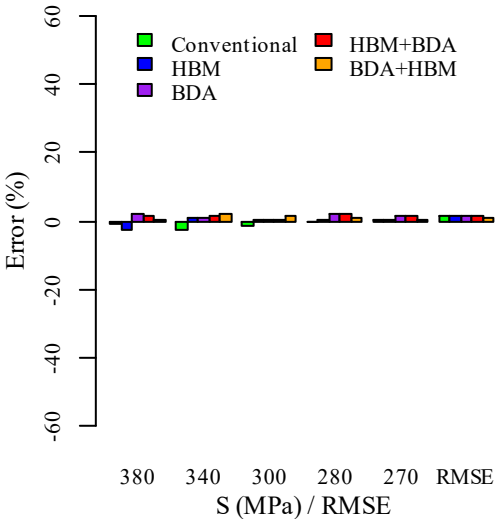
four proposed strategies can perform better than the conventional method. Among all the four strategies, in the case of sparse data, the two of the HBDA methods (i.e., HBM+BDA and BDA+HBM) have similar best performance followed by BDA, and HBM is worse than the others.

Table 2.9 Laminate Panel Data [44, 82].

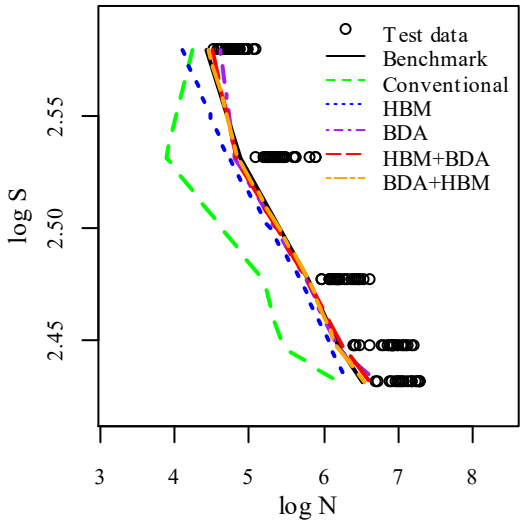
S_j	$X_j = \log(S_j)$	n_j	$Y_j = \log(N_j)$
380	2.58	25	4.534, 4.576, 4.623, 4.626, 4.683, 4.720, 4.747, 4.766, 4.790, 4.811, 4.813, 4.816, 4.848, 4.85, 4.860, 4.876, 4.889, 4.891, 4.943, 4.970, 4.973, 4.988, 4.998, 5.067, 5.088
340	2.53	25	5.099, 5.196, 5.240, 5.248, 5.254, 5.275, 5.290, 5.318, 5.326, 5.350, 5.354, 5.403, 5.407, 5.413, 5.438, 5.465, 5.478, 5.480, 5.489, 5.609, 5.624, 5.632, 5.823, 5.890, 5.900
300	2.48	25	5.980, 5.982, 6.077, 6.094, 6.097, 6.109, 6.149, 6.175, 6.181, 6.189, 6.191, 6.200, 6.215, 6.226, 6.285, 6.303, 6.337, 6.379, 6.410, 6.427, 6.466, 6.484, 6.492, 6.547, 6.635
280	2.45	23	6.416, 6.417,6.443, 6.490, 6.630, 6.778, 6.810, 6.871, 6.909, 6.929, 6.934, 6.936, 6.945, 6.945, 6.955, 7.005, 7.007, 7.047, 7.096, 7.122, 7.140, 7.186, 7.205
270	2.43	17	6.713, 6.722, 6.896, 6.913, 6.977, 7.007, 7.033, 7.063, 7.085, 7.124, 7.200, 7.204, 7.237, 7.274, 7.290, 7.295, 7.309



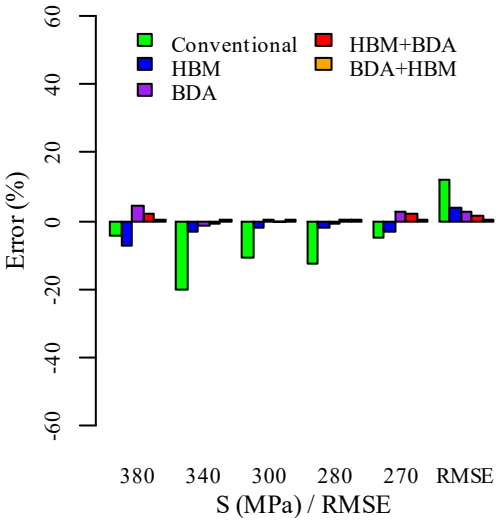
(a) Scenario 15-15-15-15-15: probabilistic *S-N* curves.



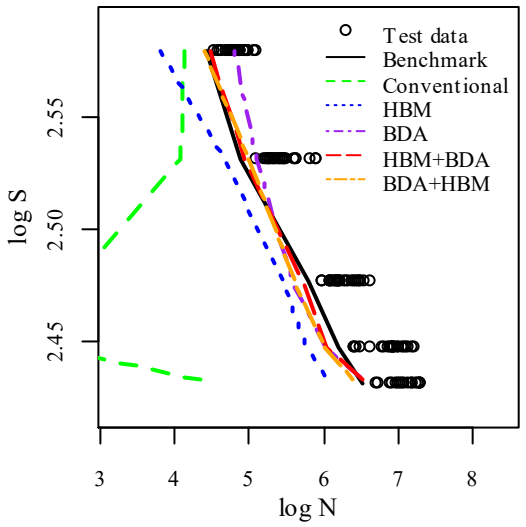
(b) Scenario 15-15-15-15-15: Errors.



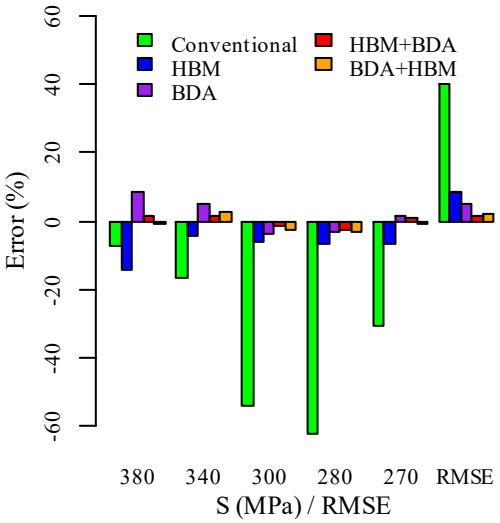
(c) Scenario 5-5-5-5-5: probabilistic *S-N* curves.



(d) Scenario 5-5-5-5-5: Errors.



(e) Scenario 3-3-3-3-3: probabilistic *S-N* curves.



(f) Scenario 3-3-3-3-3: Errors.

Fig. 2.8 Laminate Panel Data: Probabilistic *S-N* Curves with the Confidence Level of 95% and The Survivability of 99% ((a), (c), (e)) and Errors at Each Stress Levels ((b), (d), (f))

The RMSE's of different strategies with different sample sizes are plotted in Fig. 2.9. The lines show the mean values of the 50 RMSE's. The error bars are the 95% credible interval based on the 50 repeats. As the sample size decreases at each stress level, the RMSE's of all the methods increase. When the sample size is large (13 ~ 15 observations at each stress level), all the four proposed probabilistic *S-N* estimation strategies produce similar results as the conventional method. With the sample size of 4 ~ 12 observations at each stress level, the mean values of RMSE of the four proposed strategies are within 5%. The overall performance rank is HBM+BDA ≈ BDA+HBM > BDA > HBM. When the

data are sparse with 3 observations at each stress level, the mean RMSE's of HBM+BDA and BDA+HBM are less than 4% followed by BDA (about 5%) and HBM (about 6%). For scenario 2-2-2-2-2, HBM+BDA and BDA+HBM can estimate the probabilistic *S-N* curve with less than 5% mean RMSE followed by BDA (about 6%), however, HBM is not valid due to its large error. In the case of 1 observation at each stress level, all the strategies fail to give reasonable probabilistic *S-N* curves.

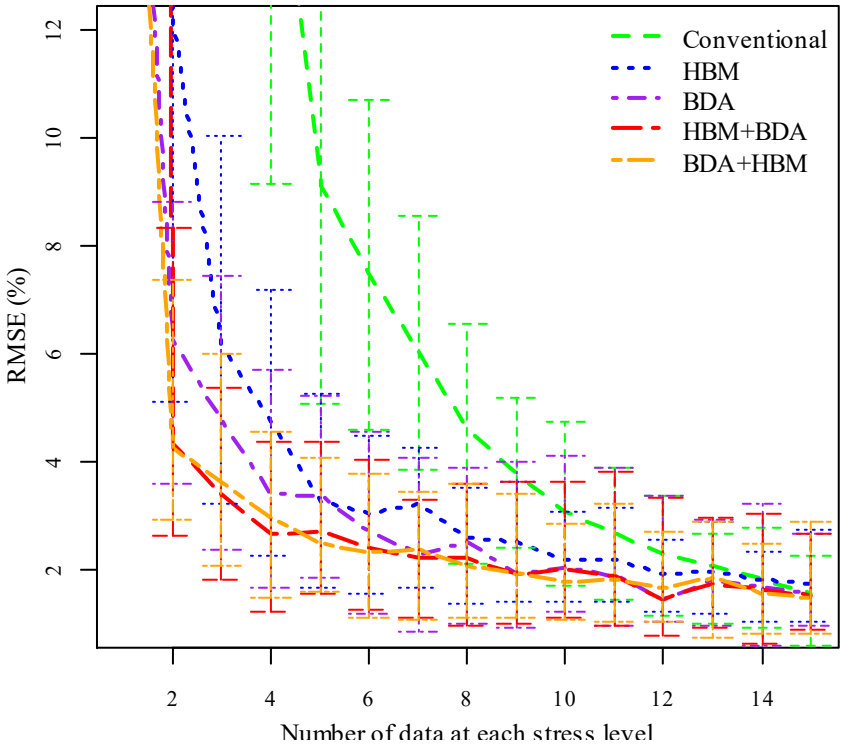


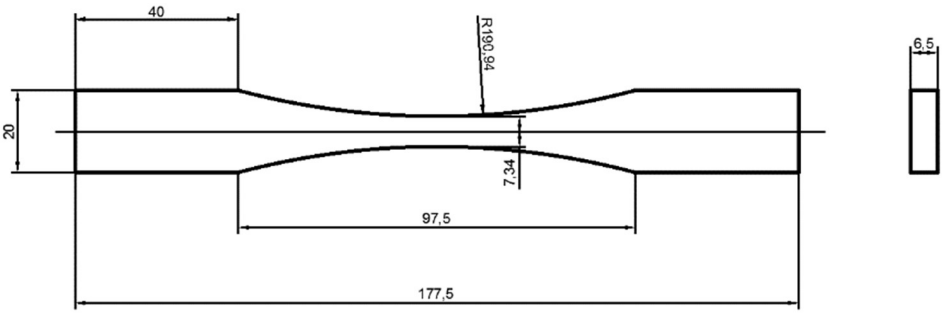
Fig. 2.9 Probabilistic RMSE's of Different Strategies with Different Sample Size.

2.3.4 Application

After being validated by data from literature references, the proposed probabilistic *S-N* estimation strategies HBDA are used for an engineering application in this section.

The specimen is manufactured from steel which was a part of the Pearl Harbor Memorial Bridge constructed in New Haven back in the 1950s. The old steel was replaced as part of the bridge improvement program. Only limited specimens are obtained for testing.

The geometry of the fatigue specimens is shown in Fig. 2.10. The design and manufacture follow ASTM E466-15: Standard Practice for Conducting Force Controlled Constant Amplitude Axial Fatigue Tests of Metallic Materials. The specimens are polished on one side of the surfaces and are kept in the original corrosion state on the other side.



(a)

(b)

(c)

Fig. 2.10 Fatigue Specimens. (a) Geometry and Size (All Dimensions in Millimeter). (b) Corroded Surface. (c) Polished Surface.

The fatigue specimens are tested at the MTS Landmark servo-hydraulic test system with constant amplitudes. The frequency is 10 Hz. The stress ratio is set to be 0.1. The tests are stopped when the specimens completely break into two pieces. The fatigue life data are shown in Table 2.10.

As shown in the laminate panel data example, one sample at a stress level is not enough for probabilistic *S-N* curve estimation. The data at the stress level of 405 and 360 MPa are used for estimation, with 5 and 6 data points respectively. The data at other stress levels with one sample are used for checking the credibility of the probabilistic *S-N* curve estimation. The probabilistic *S-N* curves with the confidence level of 95% and the 95% credible intervals are shown in Fig. 2.11. It can be seen that the two of the HBDA methods (i.e. HBM+BDA and BDA+HBM) produce similar results, while the conventional method is over-conservative.

Table 2.10 Pearl Harbor Memorial Bridge Data.

S_j	$X_j = \log(S_j)$	n_j	$Y_j = \log(N_j)$
427.5	2.631	1	5.274
405	2.607	5	5.350, 5.363, 5.397, 5.337, 5.246
382.5	2.583	1	5.547
369	2.567	1	5.627
360	2.556	6	5.524, 5.676, 5.466, 5.692, 5.626, 5.630

355.5	2.551	1	5.507
351	2.545	1	5.631
346.5	2.540	1	5.834

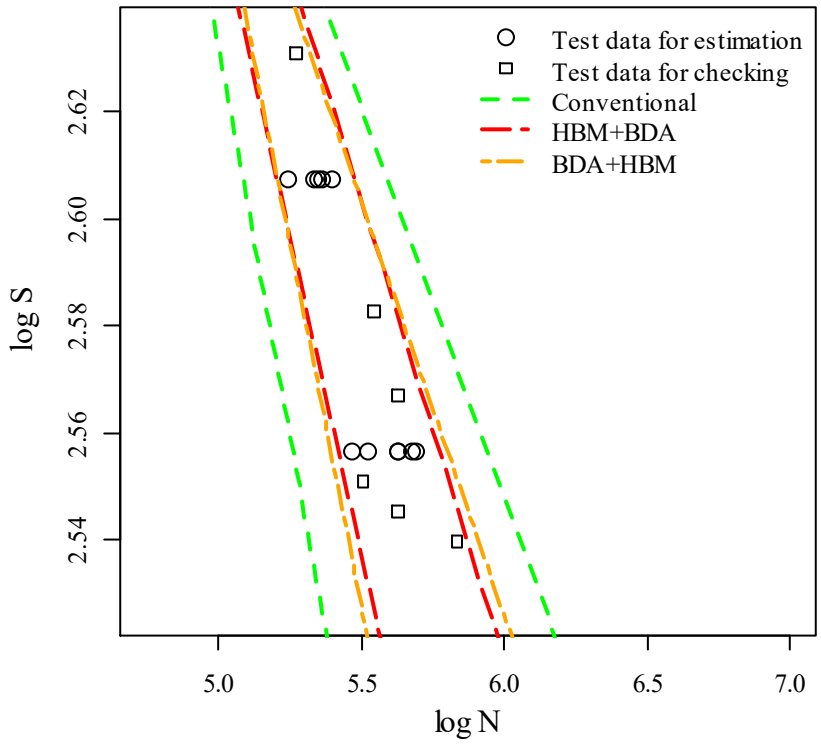


Fig. 2.11 Probabilistic *S-N* curve Estimation for Pearl Harbor Memorial Bridge Data.

2.4 Conclusions

Section 2.2 uses the Bayesian model averaging method for multimodality information fusion to estimate the probabilistic vintage pipe strength. The basic theory and practical implementation of BMA are introduced. Each model considered to be averaged is assigned a weight according to its marginal likelihood and further posterior model probability. BMA is implemented through distribution mixing. Occam's window method is used to reduce the number of models considered to improve the feasibility and efficiency

of BMA as well as keep the efficacy. The predictive performance of BMA and the single model are evaluated and compared according to the logarithmic scoring rule. Based on the current study, the following conclusions can be drawn.

1. From the analysis of full data, the best model $YS_T - YS_S = f$ (Cu, HD) has the highest probabilities. The probabilities of coefficients being non-zero show positive predictive usefulness for Cu and hardness, weak evidence for Cr, and indecisive evidence for P, Si, grain size, and volume fraction.

2. From the analysis of predictive performance, Bayesian model averaging has the advantage of improving the predictive ability by considering model uncertainty, when the number of data is insufficient to select a single best model.

Section 2.3 proposes four strategies: hierarchical Bayesian modeling (HBM), Bayesian data augmentation (BDA), posterior information from HBM used as prior information for BDA (HBM+BDA), and augmented data from BDA used by HBM (BDA+HBM) to obtain the probabilistic *S-N* curves with sparse data. HBM+BDA and BDA+HBM are together called hierarchical Bayesian data augmentation (HBDA). The theories and implementations are illustrated. The four strategies are validated using aluminum alloy 2524-T3 data, and laminate panel data. After that, the proposed methods are applied to estimate the probabilistic *S-N* curves for the Pearl Harbor Memorial Bridge. The following can be drawn from this study.

1. In the case of large data (about 15 observations per stress level), the strategies HBM, BDA, HBM+BDA, and BDA+HBM can produce similar results to those obtained from the conventional method, which shows the validity of the proposed methods with large sample size.
2. With the sparse data, the conventional method is not applicable due to large errors. In this case, the two of the HBDA methods (i.e. HBM+BDA and BDA+HBM) can estimate the probabilistic *S*-*N* curves with root mean square error less than 5%, followed by BDA and HBM. BDA is slightly better than HBM.
3. HBDA shows a large performance gain, especially when the number of the testing specimens is low (e.g., 3), and HBM+BDA and BDA+HBM have similar performance based on the investigated materials.

3 UNCERTAINTY QUANTIFICATION AND PROGNOSTICS BASED ON PHYSICS-GUIDED LEARNING

3.1 Overview

A Probabilistic Physics-guided Neural Network (PPgNN) is proposed for probabilistic fatigue *S-N* curve estimation. The proposed model overcomes the limitations in existing parametric regression models and classical machine learning models for fatigue data analysis. Compared with explicit regression-type models (such as power law fitting), the PPgNN is flexible and does not impose restrictions on function forms at different stress levels, mean stresses, or other factors. One unique benefit is that the proposed method includes the known physics/knowledge constraints in the machine learning model; the method can produce both accurate and physically consistent results compared with the classical machine learning model, such as neural network models. In addition, the PPgNN uses both failure and runout data in the training process, which encodes the runout data using a new proposed loss function, and is beneficial when compared with some existing models using only numerical point value data. A mathematical formulation is derived to include different types of physics constraints, which can deal with mean value, variance, and derivative/curvature constraints. Several data sets from open literature for fatigue *S-N* curve testing are used for model demonstration and model validation. Next, the proposed network architecture is extended to include multi-factor (e.g., mean stress, corrosion,

frequency effect, etc.) fatigue data analysis. It is shown that the proposed PPgNN can serve as a flexible and robust model for general fitting and uncertainty quantification of fatigue data. This model provides a feasible way to incorporate known physics/knowledge in neural network-based machine learning. That is achieved by properly designing the network topology and constraining the neural network's biases and weights. The benefits of the proposed physics-guided learning for fatigue data analysis are illustrated by comparing results from neural network models with and without physics guidance. The neural network model, without physics guidance, produces results contradictory to the common knowledge, such as a monotonic decrease of *S*-*N* curve slope and a monotonic increase of fatigue life variance as the stress level decreases. That problem can be avoided using the physics-guided learning model with encoded prior physics knowledge [83].

The probabilistic fatigue properties of additively manufactured (AM) Ti-6Al-4V using selective laser melted (SLM) process is analyzed considering the effects of process parameters. The Probabilistic Physics-guided Neural Network (PPgNN) is proposed for the modeling. With this developed model, both the mean and variance of the fatigue life can be learned. The PPgNN contains constraints on model parameters to obtain the probabilistic stress-life relationships (P-S-N curves) with the physics-consistent curvature and nonconstant variance. The PPgNN model is also able to be trained using the data set with missing data for more reliable predictions. Experimental fatigue data are collected

from extensive literature for AM Ti-6Al-4V in as-built and annealed conditions subjected to various process parameters (scanning speed, laser power, hatch space, layer thickness, heat temperature, heat time). The PPgNN model is validated using the experimental data. Following this, the predictive performance is compared between models training using all data (both complete and incomplete) and only complete data [84].

A Physics-guided Mixture Density Network (PgMDN) model is proposed for uncertainty quantification of regression-type analysis. It integrates a Mixture Density Network for probabilistic modeling and physics knowledge as regularizations. This model can handle arbitrary distribution of data (e.g., strongly non-Gaussian, multi-mode, and truncated distributions). The physics knowledge from parameters and their partial derivatives is used as equality/ inequality constraints. The training of physics-guided machine learning is formulated as a constrained optimization problem. To train the neural network with the commonly used backpropagation algorithm, the constrained optimization problem is transformed to an unconstrained one using a dynamic penalty function algorithm. With the physics constraints, the required training data size can be reduced and the overfitting problem can be mitigated. The applications of the PgMDN are demonstrated using a numerical example and an engineering problem for fatigue stress-life curve estimation. Some discussions are given to illustrate the effectiveness of incorporating the physics knowledge when data are sparse, the improvement of the dynamic penalty function

method compared with the static method, and the benefits achieved from the distribution mixture compared with a single Gaussian distribution.

3.2 Probabilistic Physics-guided Machine Learning for Fatigue Data Analysis

3.2.1 Introduction

The relationship between probabilistic fatigue lives of materials and applied stresses or strains is critical for the safe-life and damage tolerance design processes considering uncertainty [9, 16, 44, 85-88]. A fatigue life prediction under general random loadings requires the analysis of the probabilistic fatigue damage accumulation at various stress levels. Thus, a reliable fatigue data analysis and uncertainty quantification method play a significant role in both structural designs and analyses [89, 90].

The current attempts to establish the fatigue life–stress relationship can be categorized into explicit regression models and machine learning models. For explicit regression models, a well-known example is the Basquin relation [68], which states a linear relationship between the logarithm of the fatigue life (N) and the logarithm of the stress level (S) i.e., $\log_{10}N = \beta_0 + \beta_1 \log_{10}S$. Only two unknown parameters (slope β_1 and intercept β_0) need to be estimated when using the Basquin relation. However, the Basquin relation is only capable of describing the finite-life regime where a linear function is applicable. To characterize the effect that the fatigue life tends to be infinity when the stress level is below the fatigue limit S_0 (a low stress level), a non-linear S-N equation is proposed by

Stromeyer [91] $\log_{10}N = \beta_0 + \beta_1 \log_{10}(S - S_0)$ for $S > S_0$. Both the Basquin and the Stromeyer equation are deterministic models. To describe the fatigue life scatter dependency with applied stress levels, some statistical models have been proposed, such as the random fatigue limit model [44], the bilinear random fatigue limit model [92], and the 6-parameter random fatigue limit model [93]. The explicit regression models have the advantage that only a small number of parameters need to be estimated. However, they impose restrictions for a certain type of functional relationship between an applied loading and a fatigue life, which may or may not be true.

For the machine learning models, neural networks (NN) [94] are widely used to model the fatigue life with stress levels and other influencing factors (e.g. mean stress, environmental factors) as inputs. A deterministic relationship can be learned through the training process. The neural network is used for predicting the fatigue life of steel in a corrosive environment in Ref. [95]. In their work, overtraining and bad extrapolation problems are observed and reported. The combination of four neural networks is applied in estimating the finite-life fatigue strength and fatigue limit in Ref. [96] with inputs including stress level, notch factor, tensile strength, and yield strength. Ref. [97] models the fatigue life of multidirectional composite laminates using a neural network with one hidden layer and inputs being stress level, stress ratio, and material properties. In Ref. [98], the neural network model is fit to experimental data on the fatigue life of steel under step-

stress conditions. The constant life diagram for metallic materials is generated using a neural network in the work of Ref. [99] for high cycle fatigue. In Ref. [100], a multi-layer perceptron neural network is used for residual life prediction.

The limitations of explicit regression models are simplified representations of the reality due to incomplete knowledge, which introduce bias [101]. For the fatigue life modeling, it is well known that factors except stress levels also have a dramatic influence on the fatigue life, such as the surface roughness, temperature, and corrosion [47, 48, 53, 55]. Without a thorough understanding of the mechanics of the process, it is not an easy task to incorporate those influencing factors in explicit regression models. Due to those limitations, machine learning models are considered promising alternatives to regression models. However, a direct application of black-box machine learning models to fatigue life modeling encounters the following issues: (i) Fatigue lives show a significant scatter range even at well-controlled testing conditions. Most existing machine learning models aim to find the mean response and are not capable of flexible variance estimation at different fatigue loading conditions. (ii) The statistical relations between inputs and outputs are solely learned from the data, which may violate some commonly known physics laws or knowledge [101]; (iii) In addition, machine learning models may overfit existing data and produce unreasonable results for scenarios outside the training data range (e.g., pure extrapolation performance) [95].

To address the above-mentioned difficulties and deficiencies in the existing two types of models (explicit regression models and machine learning models), this work proposes a novel probabilistic physics-guided machine learning model for the fatigue data analysis. It aims to impose known physics/knowledge constraints on the learning process of machine learning models. The proposed model is called Probabilistic Physics-guided Neural Network (PPgNN) hereafter. Correspond to the limitations existing in both types of models in the last paragraph, the novelties of the proposed model are as follows: 1) Compared with explicit regression models, influencing factors other than the stress level are easier to be incorporated in the model by reconstructing the network architecture and the proposed model is not limited to a particular regression function in handling fatigue data under different conditions. 2) Compared with classic neural network models: (i) Fatigue life variances with respect to different stress levels can be characterized. The confidence bounds can be obtained to show the increase of the variance as the applied stress decreases. (ii) The trend of the fatigue curves is constrained according to commonly known knowledge. By imposing physics-based constraints on parameters and building a physics-based network architecture, the model can produce both accurate and physically consistent results. (iii) The overfitting issue is avoided by controlling the curvature of fitted curves from known patterns of fatigue *S*-*N* curves. Thus, fatigue life predictions outside the training data range are more robust (e.g., better extrapolation performance). (iv) What is

more, both failure and runout data can be considered in the training process. That is achieved by replacing the commonly used loss metric (e.g., mean square error) with a custom loss function.

3.2.2 Brief Review of Neural Networks

Neural networks are deep learning models. The goal of a neural network is to approximate some function f^*. For example, for regression, $y = f^*(x)$ maps an input x to a numerical y. A neural network defines a mapping $y = f(x; \theta)$ and learns the value of the parameters θ that result in the best function approximation. The network is called neural because it is loosely inspired by neuroscience [5].

A typical single hidden layer neural network shown in Fig. 3.1 will be used as a demonstration for the basic knowledge of neural networks and to give the preliminary before the probability learning and imposing the physics guidance. This example consists of three layers, each of which has a particular number of nodes (neurons). The first layer is called the input layer and consists of several sensory neurons (neurons that make no processing, they just sense incoming signals and pass them to the next layer). Note that this figure is for a two-dimensional input x. The last layer (the one that produces the final results of the network) is called the output layer and consists of several computational neurons. There is one output in this example. The layer between the input and output layer is called the hidden layer with a fixed number of computational neurons [97]. The layers

are fully connected in this example. Typically, the number of hidden units is somewhere in the range of 5 to 100, with the number increasing as the number of inputs and the number of training cases increase. Choices of the number of hidden layers are guided by background knowledge and experimentation [6]. This neural network is called feedforward because the information flows from the function being evaluated from x, through the intermediate computations used to define f, and finally to the output y [5].

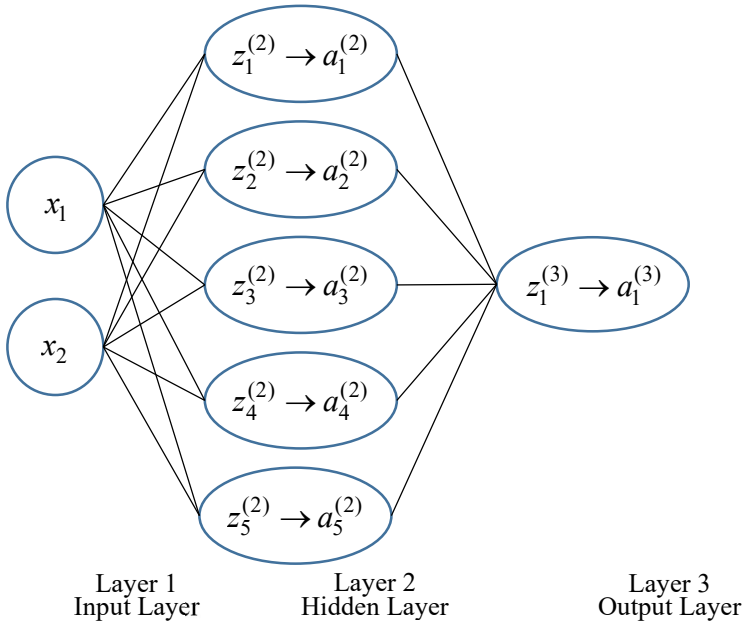


Fig. 3.1 An Example of A Single Hidden Layer Neural Network.

The mathematical expression for the neuron within each layer for Fig. 3.1 can be given in the following equations [102].

$$z_k^{(l)} = b_k^{(l-1)} + \sum_{j=1}^{p_{l-1}} w_{kj}^{(l-1)} a_j^{(l-1)} \quad l = 2,...,L\,, \tag{3.1}$$

$$a_k^{(l)} = g_k^{(l)}\left(z_k^{(l)}\right) \quad k = 1,2,...,p_l\,. \tag{3.2}$$

where l from 2 to L is the index of the layer with $l = 1$ being the input layer and $l = L$ being the output layer. p_l is the number of neurons at the lth layer. For the input layer,

$$a_k^{(1)} = x_k \quad k = 1,...,p_1\,. \tag{3.3}$$

$g_k^{(l)}(\cdot)$ is the activation function associated with the kth neuron at the lth layer. The activation function of a neuron defines the output of that neuron given the inputs from the previous layer. The intercepts and the coefficients of Eq. (3.1) are called biases and weights, respectively. The biases and weights are obtained by training the network given the data. The loss function is minimized during training. It is a quantity that represents a measure of success for the task [103]. The stochastic gradient descent method is employed for solving the optimization problem. Stochastic gradient descent computes the gradient using subsets of the data called minibatches. The weights and biases are updated minibatch by minibatch in one cycle until the entire data set is gone through and then followed by another cycle. Each cycle is called an epoch.

Fig. 3.1 shows a standard neural network architecture, which is not necessarily the most suitable one for a specific problem. The term neural network architecture refers to the arrangement of neurons into layers and the connection patterns between layers, activation

functions, and learning methods. The neural network model and the architecture of a neural network determine how a network transforms its inputs into outputs [104]. To extend the standard neural network for the fatigue data analysis, an appropriate architecture named Probabilistic Physics-guided Neural Network (PPgNN) is proposed and illustrated in the next sections. Two major characteristics of the proposed architectures for fatigue life data modeling are as follows. 1). The proposed architecture is designed to account for not only the change of the mean of fatigue life with respect to the influence factors (e.g., the stress or strain) but also the change of the scatter at different stress levels. This is achieved by assigning two neurons at the output layer rather than one. They are the mean and standard deviation of fatigue lives, respectively. Also, a custom loss function is proposed to consider both failure data and censored data (runouts). 2). The proposed neural network produces physically consistent results to account for the life-stress curvature and nonconstant variability. The neural network is trained through a constrained optimization process. The constraints are imposed to guide the neural network to follow existing physical knowledge during training. The above two main characteristics show the probabilistic and physic-guidance features of the proposed PPgNN, respectively. The proposed methodology is described in detail in the following two sub-sections. Firstly, the fatigue life data are modeled as the function of a single factor, i.e., stress or strain. Next, the proposed PPgNN is extended to incorporate other fatigue life-influencing factors, e.g., mean stress.

3.2.3 Probabilistic Modeling of Fatigue Data in Neural Networks

The architecture of the probabilistic neural network is shown in Fig. 3.2. Three layers are included in this network: the input layer, one hidden layer, and the output layer. The input layer consists of stress or strain level (S), logarithmic of fatigue life ($log\ (N)$), and an index to indicate if a datum is a failure or runout. There are 5 neurons in the hidden layer. The number of neurons in the hidden layer can be adjusted and is not necessarily to be 5 as shown in Fig. 3.2. This number is chosen to be 5 because of the consideration that the fatigue S-N curve is not expected to have complicated curve shapes and only one factor, stress (strain), is considered. Thus, the minimum number is chosen according to the recommendation (5 – 100 neurons per hidden layer [6]). The probabilistic aspect of the neural network is reflected in the output layer. Unlike the regular neural network that only learns the mean from the collected distributed data, there are two output neurons in the output layer of the proposed network, both the mean and standard deviation.

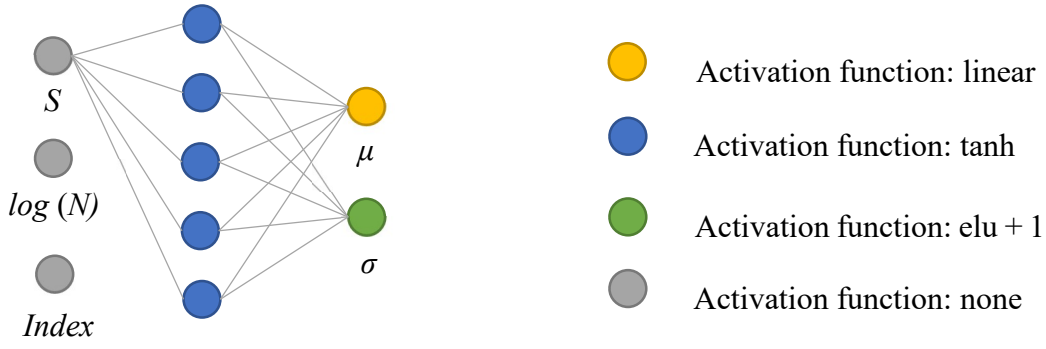


Fig. 3.2 The Architecture of the Probabilistic Neural Network.

Different colors of neurons in Fig. 3.2 indicate specific activation functions. For the input layer, no activation functions are used by the definition of the neural network. The hyperbolic tangent activation function (named tanh) is used for the neurons at the hidden layer. The expression of the tanh activation function is

$$g(z)=\frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}. \tag{3.4}$$

The tanh activation function is commonly used for the hidden layer neurons and gives good results in this work. The mean (μ) in the output layer has the linear (i.e., identity) activation function which is usually used for outputs in the regression problem,

$$g(z)=z. \tag{3.5}$$

Due to the fact that the standard deviation has a non-negative value, the exponential linear unit (elu) activation function is selected and modified by adding 1 as following

$$g(z)=\begin{cases} z+1 & z\geq 0 \\ e^{z} & z<0 \end{cases}. \tag{3.6}$$

This is named elu+1 in this work and has the property that it can transform an arbitrary number to a non-negative value. Other activation functions that have this property also work, for example, the absolute value function [105]. Elu+1 seems to work well in this work as the validation later in this work shows very good results.

In the neural network architecture in Fig. 3.2, the inputs are *S*, log (*N*), and *Index*. Only the *S* input is passed to the next layers. The inputs log (*N*) and *Index* are not connected with the next layer, and the data for these two inputs are used in the custom loss function for measuring the performance of the model. The loss function is customized to be able to estimate both the mean and standard deviation using all data including failures and runouts. The loss function is defined as the negative logarithm of the likelihood [106]. In other words, the neural network is trained to maximize the likelihood. The new loss function in this work is proposed as

$$L = -\sum_{i=1}^{n}\left\{\delta_i \log\left[f\left(\log(N)\,|\,\mu,\sigma\right)\right] + \left(1-\delta_i\right)\log\left[1-F\left(\log(N)\,|\,\mu,\sigma\right)\right]\right\}. \tag{3.7}$$

where δ_i is the index of a failure or runout, and

$$\delta_i = \begin{cases} 1 & \text{for failure} \\ 0 & \text{for runout} \end{cases}.$$

f (log (N) | μ, σ) and F (log (N) | μ, σ) are the probability density function and cumulative distribution function respectively with the location parameter μ and scale parameter σ. The normal distribution or smallest extreme value distribution can be adopted. The neural network is used for regression in this work. The probabilistic characteristic is reflected in the construction of the loss function. The negative log-likelihood of the data is used for the loss function. That is, the likelihood is maximized after training the neural network. A form of distribution is presumed (normal distribution in this work) for the loss

function. A normal distribution contains two unknown parameters: the mean μ, and the standard deviation σ. These two parameters are the outputs. The normal distribution adopted is a reasonable distribution for the log fatigue life. However, the distribution form can be changed according to applications in different types of data. For example, if investigated data follow three-parameter Weibull distribution, the number of outputs will be 3: shape, scale, and location parameter.

The architecture shown in Fig. 3.2 with the custom loss function is feasible to obtain the mean curve and confidence bounds of the fatigue life data versus the stress (strain). The curves are obtained with the maximum likelihood. However, the best-estimated curves that are learned solely from the data may present physically inconsistent curvature within the range of observed data and will perform very badly for extrapolation. To overcome these drawbacks, physics knowledge is imposed during the training process to guide the neural network to produce physically reasonable results. In the case of the fatigue data analysis, the desired features of *S*-*N* curves are expected to show the increased scatter with the decrease of stress level and an infinity or specified large number of cycles at low stress levels. The proposed techniques for imposing the physics guidance are described in the following subsection.

3.2.4 Physics-guided Machine Learning

There are two main considerations in modeling the fatigue life with the applied stress or strain. First, the standard deviation of the fatigue life, in general, increases as the stress (strain) decreases. Second, the curvature of the fatigue curve decreases as the stress decreases and shows an asymptotic behavior near the fatigue limit or very long life if the material does not have an apparent fatigue limit [44]. These two aspects are the physics knowledge for guiding the machine learning process to obtain the results with those desired characteristics.

The physics knowledge is incorporated to the neural network using imposing appropriate constraints on weights, biases, or both. The derivations of different types of constraints are as follows. According to the neural network architecture in Fig. 3.2 and mathematical expressions in Eqs. (3.1) – (3.6), with the input stress (strain) level S, the kth neurons in the hidden layer (the second layer) have the form of

$$z_k^{(2)} = b_k^{(1)} + w_{k1}^{(1)} S \quad \rightarrow \tag{3.8}$$

$$a_k^{(2)} = g_k^{(2)}\left(z_k^{(2)}\right) \tag{3.9}$$

with $g_k^{(2)}(z)$ being the tanh activation function shown in Eq. (3.4). The outputs μ and σ in the output layer have the forms of

$$z_1^{(3)} = b_1^{(2)} + \sum_j w_{1j}^{(2)} a_j^{(2)} \quad \rightarrow \tag{3.10}$$

$$\mu = g_1^{(3)}\left(z_1^{(3)}\right) \tag{3.11}$$

with $g_1^{(3)}(z)$ being the linear activation function shown in Eq. (3.5), and

$$z_2^{(3)} = b_2^{(2)} + \sum_j w_{2j}^{(2)} a_j^{(2)} \quad \rightarrow \tag{3.12}$$

$$\sigma = g_2^{(3)}\left(z_2^{(3)}\right) \tag{3.13}$$

with $g_2^{(3)}(z)$ being the elu+1 activation function shown in Eq. (3.6).

The derivations of different types of constraints are as follows. From Eq. (3.8) – (3.13), it can be seen that the outputs μ and σ are the functions of the input S. The first and second derivatives of μ with respect to S are

$$\begin{aligned}
\frac{d\mu}{dS} &= \frac{d\mu}{dz_1^{(3)}}\left(\sum_j \frac{dz_1^{(3)}}{da_j^{(2)}}\frac{da_j^{(2)}}{dz_j^{(2)}}\frac{dz_j^{(2)}}{dS}\right) \\
&= 1\times\left(\sum_j w_{1j}^{(2)}\frac{4}{\left(e^{z_j^{(2)}}+e^{-z_j^{(2)}}\right)^2}w_{j1}^{(1)}\right) \\
&= \sum_j w_{j1}^{(1)} w_{1j}^{(2)}\frac{4}{\left(e^{z_j^{(2)}}+e^{-z_j^{(2)}}\right)^2}
\end{aligned} \tag{3.14}$$

and

$$\frac{d^2\mu}{dS^2} = \sum_j w_{j1}^{(1)} w_{1j}^{(2)} \left(\frac{4}{\left(e^{z_j^{(2)}} + e^{-z_j^{(2)}} \right)^2} \right)' \frac{dz_j^{(2)}}{dS}$$
$$= \sum_j -\left[w_{j1}^{(1)} \right]^2 w_{1j}^{(2)} \frac{8\left(e^{z_j^{(2)}} - e^{-z_j^{(2)}} \right)}{\left(e^{z_j^{(2)}} + e^{-z_j^{(2)}} \right)^3} \quad , \tag{3.15}$$

respectively. The first and second derivatives of σ with respect to S are

$$\frac{d\sigma}{dS} = \frac{d\sigma}{dz_2^{(3)}} \left(\sum_j \frac{dz_2^{(3)}}{da_j^{(2)}} \frac{da_j^{(2)}}{dz_j^{(2)}} \frac{dz_j^{(2)}}{dS} \right)$$
$$= \begin{cases} 1 \times \left(\sum_j w_{2j}^{(2)} \frac{4}{\left(e^{z_j^{(2)}} + e^{-z_j^{(2)}} \right)^2} w_{j1}^{(1)} \right) & z_2^{(3)} \geq 0 \\ e^{z_2^{(3)}} \left(\sum_j w_{2j}^{(2)} \frac{4}{\left(e^{z_j^{(2)}} + e^{-z_j^{(2)}} \right)^2} w_{j1}^{(1)} \right) & z_2^{(3)} < 0 \end{cases}$$
$$= \begin{cases} \sum_j 4 w_{j1}^{(1)} w_{2j}^{(2)} \frac{1}{\left(e^{z_j^{(2)}} + e^{-z_j^{(2)}} \right)^2} & z_2^{(3)} \geq 0 \\ e^{z_2^{(3)}} \sum_j 4 w_{j1}^{(1)} w_{2j}^{(2)} \frac{1}{\left(e^{z_j^{(2)}} + e^{-z_j^{(2)}} \right)^2} & z_2^{(3)} < 0 \end{cases} \tag{3.16}$$
$$= \sum_j -\left[w_{j1}^{(1)} \right]^2 w_{1j}^{(2)} \frac{8\left(e^{z_j^{(2)}} - e^{-z_j^{(2)}} \right)}{\left(e^{z_j^{(2)}} + e^{-z_j^{(2)}} \right)^3}$$

and

$$\frac{d^2\sigma}{dS^2}=\begin{cases}\displaystyle\sum_j 4w_{j1}^{(1)}w_{2j}^{(2)}\left(\frac{1}{\left(e^{z_j^{(2)}}+e^{-z_j^{(2)}}\right)^2}\right)'\frac{dz_j^{(2)}}{dS} & z_2^{(3)}\geq 0\\ \left[e^{z_2^{(3)}}\displaystyle\sum_j\frac{dz_2^{(3)}}{da_j^{(2)}}\frac{da_j^{(2)}}{dz_j^{(2)}}\frac{dz_j^{(2)}}{dS}\right]\sum_j 4w_{j1}^{(1)}w_{2j}^{(2)}\frac{1}{\left(e^{z_j^{(2)}}+e^{-z_j^{(2)}}\right)^2}+ & \\ \qquad e^{z_2^{(3)}}\displaystyle\sum_j 4w_{j1}^{(1)}w_{2j}^{(2)}\left(\frac{1}{\left(e^{z_j^{(2)}}+e^{-z_j^{(2)}}\right)^2}\right)'\frac{dz_j^{(2)}}{dS} & z_2^{(3)}<0\end{cases}$$

$$=\begin{cases}\displaystyle\sum_j -8\left[w_{j1}^{(1)}\right]^2 w_{2j}^{(2)}\frac{\left(e^{z_j^{(2)}}-e^{-z_j^{(2)}}\right)}{\left(e^{z_j^{(2)}}+e^{-z_j^{(2)}}\right)^3} & z_2^{(3)}\geq 0\\ e^{z_2^{(3)}}\left[\displaystyle\sum_j 4w_{j1}^{(1)}w_{2j}^{(2)}\frac{1}{\left(e^{z_j^{(2)}}+e^{-z_j^{(2)}}\right)^2}\right]^2+ & \\ \qquad e^{z_2^{(3)}}\displaystyle\sum_j -8\left[w_{j1}^{(1)}\right]^2 w_{2j}^{(2)}\frac{\left(e^{z_j^{(2)}}-e^{-z_j^{(2)}}\right)}{\left(e^{z_j^{(2)}}+e^{-z_j^{(2)}}\right)^3} & z_2^{(3)}<0\end{cases}, \tag{3.17}$$

respectively.

For the first desired characteristic that the standard deviation of the fatigue life increases as the stress (strain) decreases, it requires that

$$\frac{d\sigma}{dS}\leq 0\,, \tag{3.18}$$

From Eq. (3.16), the first derivative of σ is the summation of several terms. In this work, each term is constrained to be non-positive to make sure the sum is non-positive.

This is a somewhat stronger constraint than the original requirement, but it is easier to implement and ensure the final results are consistent with the known physics. Thus,

$$w_{j1}^{(1)} w_{2j}^{(2)} \le 0, \tag{3.19}$$

In other words,

$$w_{j1}^{(1)} \ge 0, \quad w_{2j}^{(2)} \le 0, \tag{3.20}$$

or

$$w_{j1}^{(1)} \le 0, \quad w_{2j}^{(2)} \ge 0, \tag{3.21}$$

For the second desired characteristic that the curvature of the fatigue curve decreases as the stress (strain) decreases, it can be satisfied if

$$\frac{d^2 \mu}{dS^2} \ge 0, \tag{3.22}$$

Similar to the constraints for the standard deviation, each term of Eq. (3.15) is restricted to be non-negative, which gives the following constraints

$$w_{1j}^{(2)} \le 0, \quad z_j^{(2)} \ge 0, \tag{3.23}$$

or

$$w_{1j}^{(2)} \ge 0, \quad z_j^{(2)} \le 0, \tag{3.24}$$

To have non-negative or non-positive values for $z_j^{(2)}$, from Eq. (3.8), since the stress (strain) is always positive, if

$$b_j^{(1)} \geq 0, \quad w_{j1}^{(1)} \geq 0\,, \tag{3.25}$$

then

$$z_j^{(2)} \geq 0\,, \tag{3.26}$$

can be guaranteed. And similarly, if

$$b_j^{(1)} \leq 0, \quad w_{j1}^{(1)} \leq 0\,, \tag{3.27}$$

then

$$z_j^{(2)} \leq 0\,, \tag{3.28}$$

Thus, for the curvature to be physically consistent, either set of the following constraints needs to be satisfied

$$b_j^{(1)} \geq 0, \quad w_{j1}^{(1)} \geq 0, \quad w_{1j}^{(2)} \leq 0\,, \tag{3.29}$$

or

$$b_j^{(1)} \leq 0, \quad w_{j1}^{(1)} \leq 0, \quad w_{1j}^{(2)} \geq 0\,, \tag{3.30}$$

From Eq. (3.15) and (3.17), if the second derivative of μ is constrained to be non-negative, the second derivative of σ will also have the non-negative value. Experimental data for fatigue curves do not show the evidence for the increase rate of the standard deviation with the decrease of the stress (second derivative of σ). To avoid the correlation between the second derivatives of μ and σ, the architecture is redesigned to separate the neurons that are connected to μ and σ in the hidden layer. The updated architecture is shown

in Fig. 3.3. The connection lines are with different colors to show different types of constraints corresponding to constraints (3.20) and (3.29) (or constraints (3.21) and (3.30)).

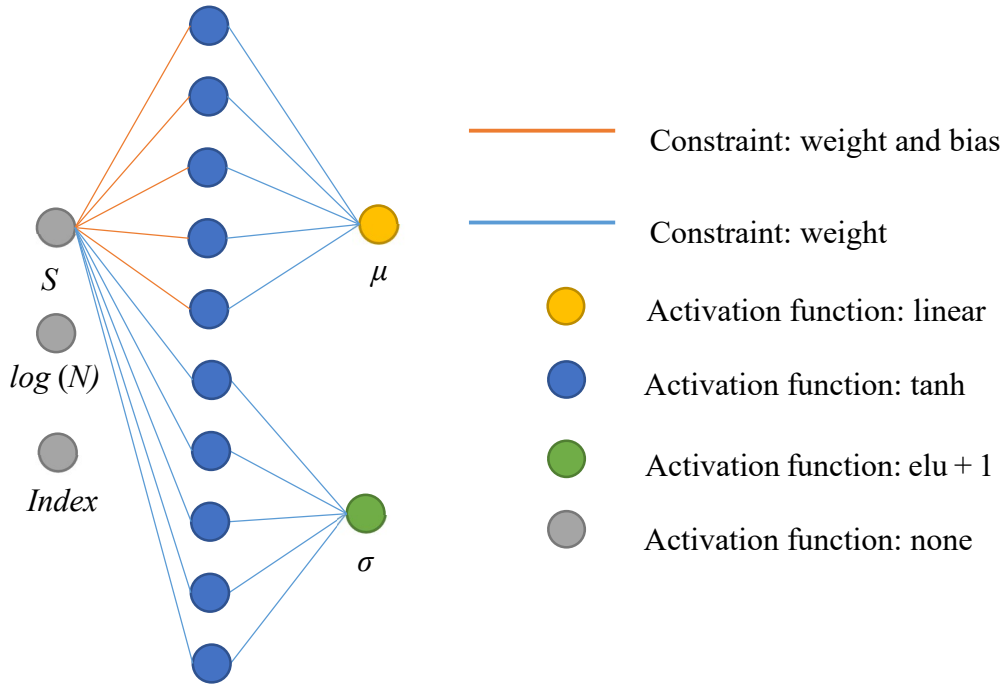


Fig. 3.3 The Architecture of Probabilistic Neural Network with Weight and Bias Constraints.

The proposed neural network with the weight or bias constraints can produce the results which satisfy that the curvature decreases as the stress decreases. A typical *S*-*N* curve shows an asymptotic behavior near the fatigue limit. This means that the second derivative of μ tends to be zero as the stress (strain) decreases approaching the fatigue limit. However, due to the overly restrictive constraints for constraint (3.22) (i.e., each term is constrained to be non-negative to guarantee the sum to be non-negative), the minimum

value of the second derivative of μ is a positive value. To relax the curvature, the architecture is modified by adding another layer with a relaxed μ_r and connecting μ and σ to μ_r. The μ_r and σ are the final outputs. The modified architecture is shown in Fig. 3.4 and is used as the proposed architecture of Probabilistic Physics-guided Neural Network for single factor (S). The weights of the third layer are constrained to be non-negative in order not to change the constraints at previous layers. The second derivative of μ_r is the sum of the second derivatives of μ and σ. Since the second derivative of σ is unconstrained, it can have either a positive or negative value, which provides the relaxation on the second derivative of μ_r.

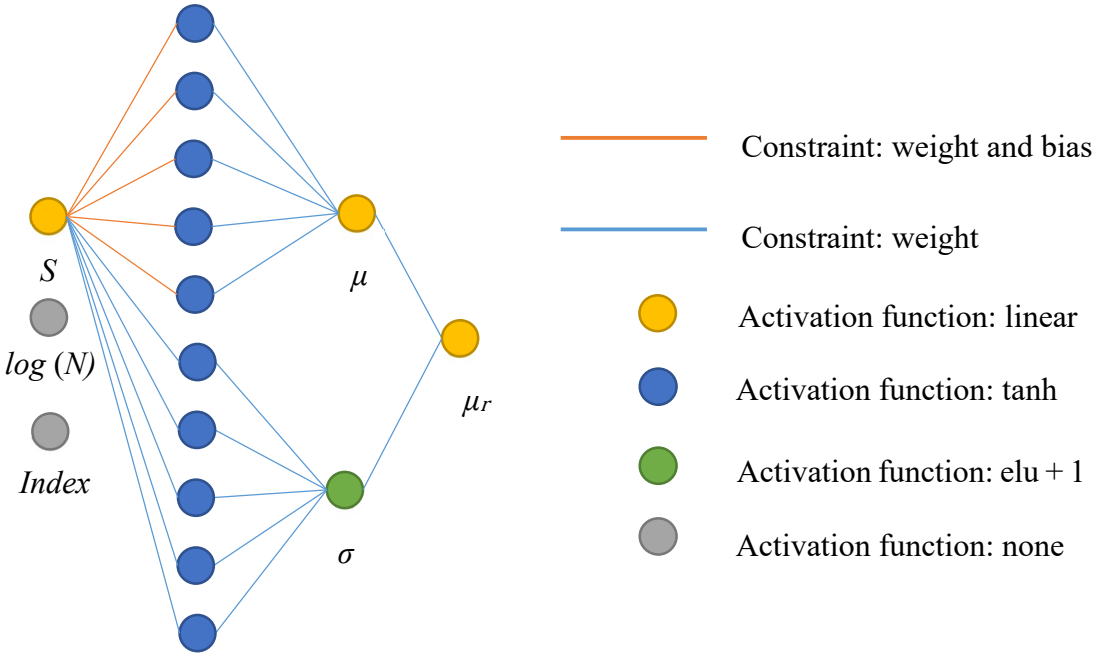


Fig. 3.4 The Architecture of Probabilistic Neural Network with Stress (Strain) Input.

The architecture shown in Fig. 3.4 is for the single factor case, i.e., stress (or strain) vs. life. An advantage of the proposed physics-guided machine learning model compared with the classical regression type model is the flexibility and scalability to consider the influence of other factors. Other inputs can be added to the network according to the known physical knowledge. An example is to consider the mean stress (S_m) effect. For a given fatigue load range, a tensile mean stress has a detrimental effect on the fatigue strength, whereas, in general, a compressive mean stress has a beneficial effect [107]. The mean fatigue life increases with the reduction of mean stresses. The influence of mean stress on the scatter is unclear. Thus, the first derivative of μ with respect to the mean stress is non-positive, which gives constraints analogous to constraints (3.20) and (3.21). Since no physics knowledge is given to restrict the mean stress – variance relationship, no constraint is added, and the relationship is obtained solely from the data training. The architecture to consider the mean stress is given in Fig. 3.5.

In the next section (Experimental validation), various data sets are used to validate the proposed PPgNN architecture in Fig. 3.4 for the single factor experiment and Fig. 3.5 for the multi-factor experiment. Following the experimental validation, in Section Discussions, the necessities for the physics guidance are described, and the comparison between results from the neural network without and with physics guidance is discussed.

3.2.5 Experimental Validation

In this section, extensive evaluations of the proposed PPgNN are conducted using various fatigue data sets from open literature for both single factor and multi-factor cases.

The learning algorithm's parameters are kept the same for all the following experiments with different data sets and are shown in Table 3.1.

Table 3.1 Learning Algorithm's Parameters.

Epochs	500 for single-factor experiment; 1000 for multi-factor experiment
Minibatch size	1
Learning rate	0.001

Six different fatigue data sets are collected, which cover a wide range of metallic and composite materials. Table 3.2 is a summary of the collected data.

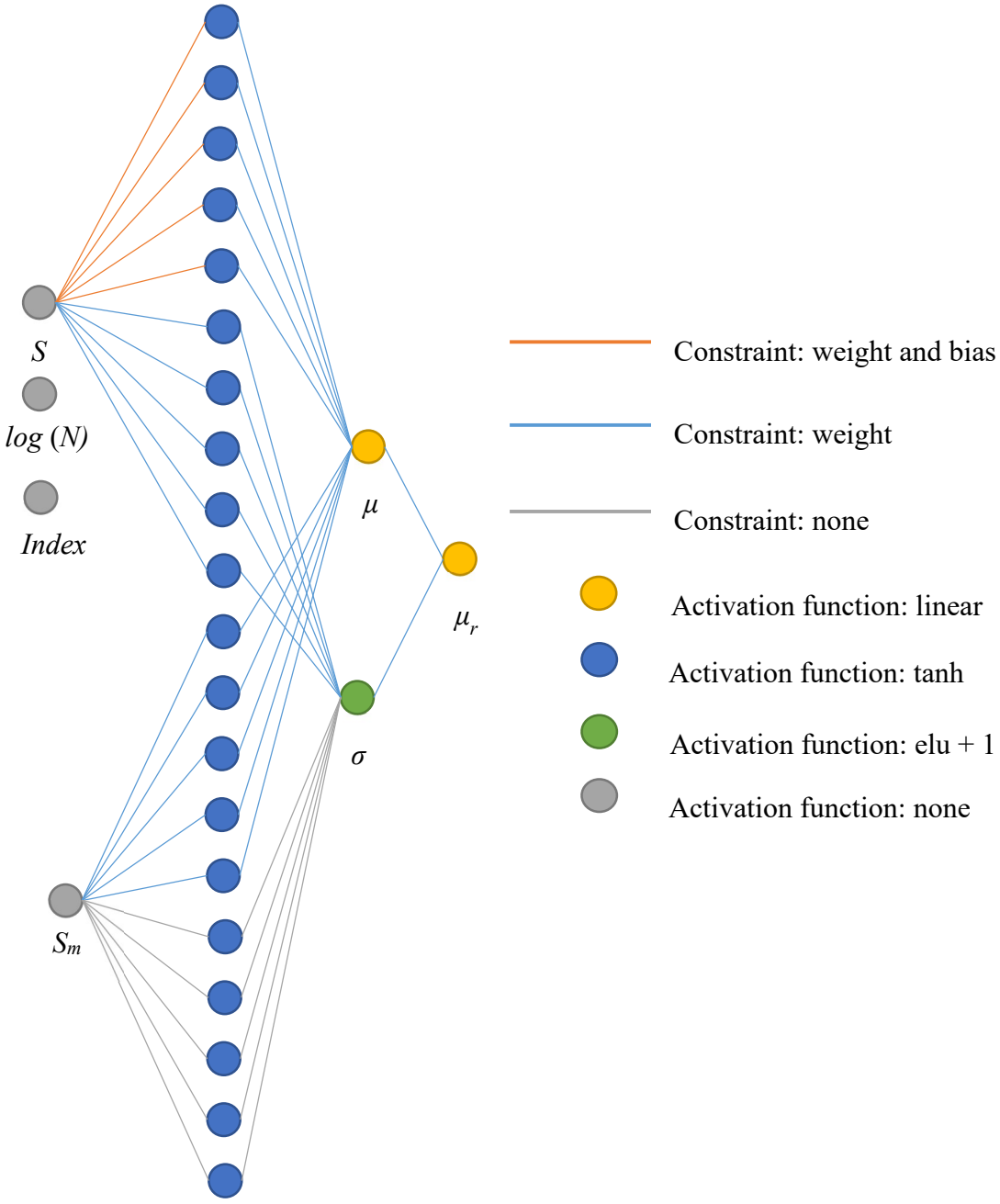


Fig. 3.5 The Architecture of Probabilistic Neural Network with Multiple Inputs.

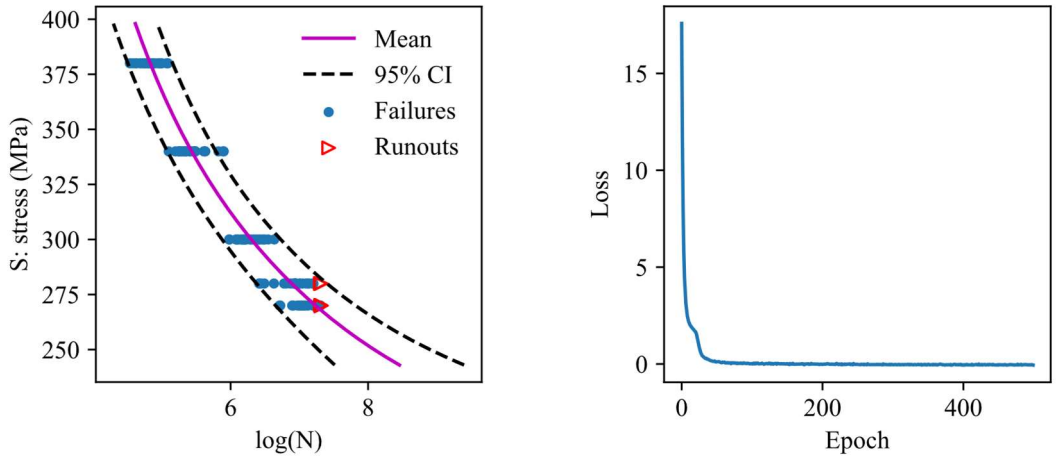
Table 3.2 Summary of the Data Set Used for Validation.

Material	Reference	Sample size	No. of Runouts
Carbon eight-harness-satin/epoxy laminate	[44, 82]	125	10
Nickel-base superalloy	[44, 108]	246	4
Annealed aluminum	[108]	200	NA
Steel	[108]	75	10
Al 2024-T4	[108]	252	NA
Al 7075-T651	[109]	131	8

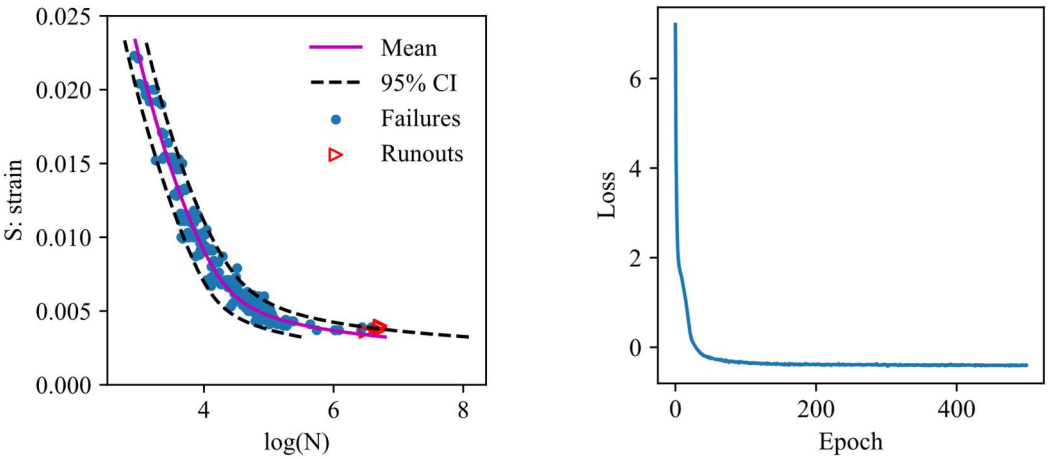
3.2.5.1 Single Factor PPgNN Validation

For the single-factor experiment, all probabilistic *S-N* curves using the Probabilistic Physics-guided Neural Network (PPgNN) are shown in Fig. 3.6 for various materials. All left column figures are for the mean and 95% confidence bounds plot with experimental data. The solid lines and dash lines represent the mean *S-N* curves and 95% confidence intervals (CI), respectively. The *x*-axis is the fatigue life and the *y*-axis is the stress/strain. "Runout" data are highlighted in the figures. All right column figures are training convergence plots using the proposed loss function. The *x*-axis is the Epoch number and the *y*-axis is the loss value. Very good convergence behavior is observed using the proposed loss function.

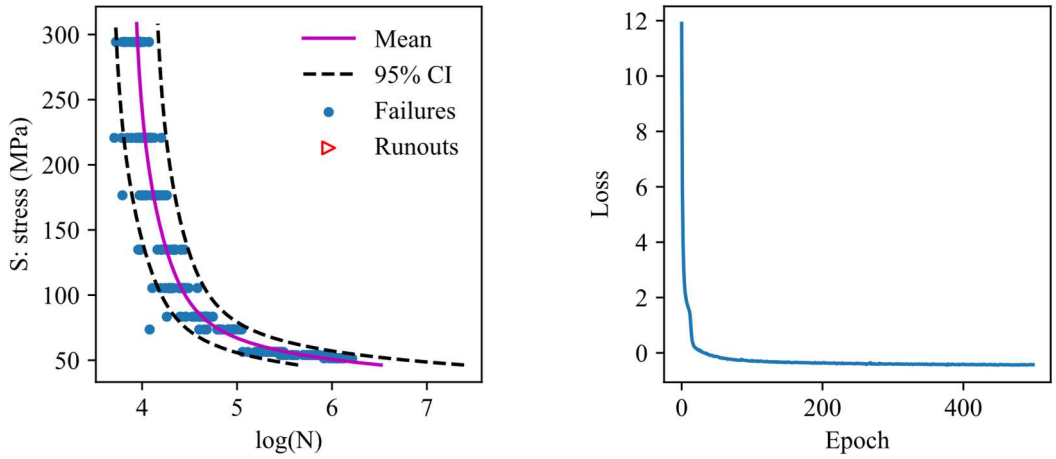
By using the PPgNN model, the desired results (i.e., not violating known physics about the fatigue performance of material) can be achieved. With the physics constraints in the neural network, the obtained *P*-*S*-*N* curves show consistent physics results. First, the standard deviation of fatigue life increases as the stress (strain) decreases. Second, the curvature of the fatigue curve decreases as the stress decreases and shows an asymptotic behavior near the fatigue limit or very long life if the material does not have an apparent fatigue limit. PPgNN model is shown to be a model suitable for the fatigue life data fitting for a variety of different materials. Also, the PPgNN is not limited to a particular stress (strain) region. For example, the nickel-base superalloy data (Fig. 3.6 (b)) was used for *P*-*S*-*N* curve fitting in Ref. [44] using the random fatigue limit model which belongs to the explicit regression model category. This model did not fit data well in the high-strain region (strain above 0.007) [44]. A possible reason is that this type of model has a fixed function form which may not be appropriate for all strain regions. This issue does not exist using the proposed PPgNN model due to the flexibility of the neural network. Thus, both low-strain and high-strain data can be used together for the *P*-*S*-*N* curve fitting. In addition, both deterministic and probabilistic fatigue properties can be obtained simultaneously using the same framework. Not only the mean *S*-*N* curves but also confidence bounds can be estimated from the data. It can be seen that almost all the data points are located within the 95% CI.



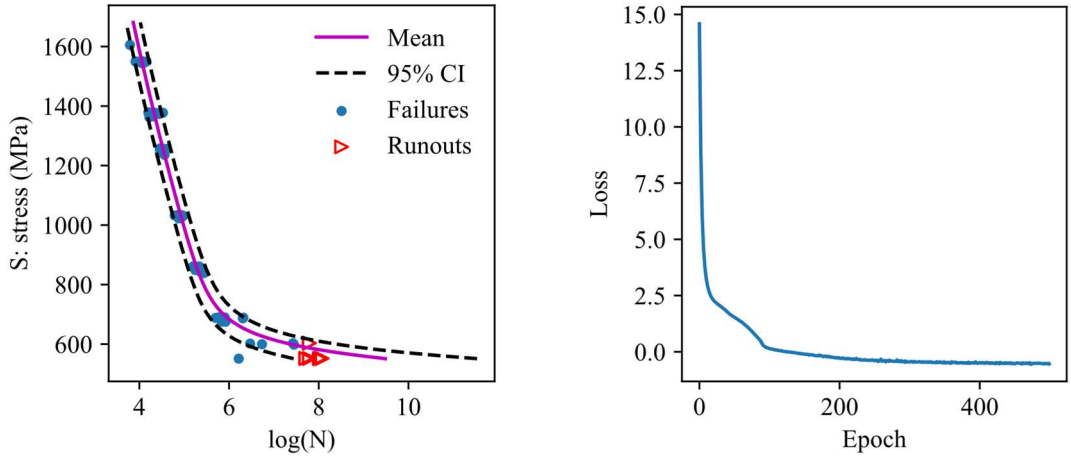
(a) Carbon eight-harness-satin/epoxy laminate.



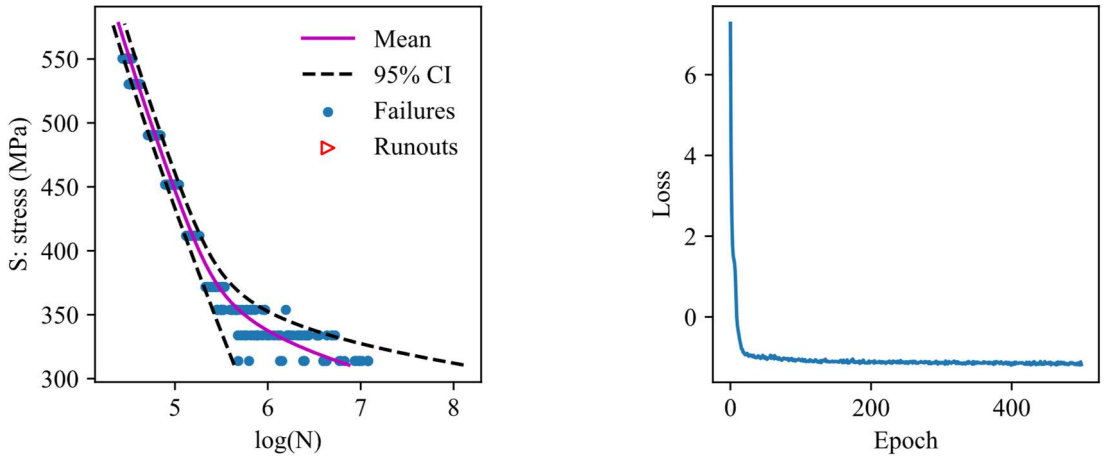
(b) Nickel-base superalloy.



(c) Annealed aluminum.



(d) Steel.



(e) Al 2024-T4.

Fig. 3.6 Validation of PPgNN Using Single Factor Experimental Data for Various Materials.

3.2.5.2 Multi-factor PPgNN Validation

One advantage of the Probabilistic Physics-guided Neural Network (PPgNN) compared with explicit regression models is that it is easier to incorporate other influencing factors other than stress (strain). This experiment is to show the effectiveness of the PPgNN shown in Fig. 3.5 with multi-factor inputs.

The data set used contains strain level, mean stress, and fatigue life [109]. The material is 7075-T651 aluminum alloy. The fatigue testing is conducted under uniaxial compression-compression, tension-compression, and tension-tension fatigue loading

conditions. The mean stress can have a negative value, zero, or positive value. A total of 131 data are used for fitting among which 8 data are right-censored (runouts).

The architecture proposed in Fig. 3.5 is used for the curve fitting. The results are shown in Fig. 3.7 Validation of PPgNN Using Multi-factor Experimental Data with Mean Stress Effect.. Fig. 3.7 Validation of PPgNN Using Multi-factor Experimental Data with Mean Stress Effect. (a), (b) and (c) are the probabilistic *S*-*N* curves with 95% confidence intervals for S_m = -250 MPa, 0 MPa, and 150 MPa, respectively. The testing data corresponding to each mean stress case are also plotted. It can be seen that for each of the three mean stress cases, almost all data points are located within the 95% confidence interval. Again, the results show that the standard deviation of fatigue life increases, and the curvature of the fatigue curve decreases as the strain decreases. Fig. 3.7 Validation of PPgNN Using Multi-factor Experimental Data with Mean Stress Effect. (d) is the model training history, the loss function value vs. the epoch.

Ref. [110] extended the random fatigue limit model [44] to be able to consider influencing factors other than stress (strain) level, i.e., multi-factor fatigue data analysis. This is a type of explicit regression model, and the function form needs to be known prior to data fitting. Thus, to incorporate multiple factors in the model, data need to be pre-analyzed to find a reasonable functional form. However, using the PPgNN model, those pre-analyses are not mandatory. The neural network in the PPgNN model learns the

function from the data, and constraints in PPgNN ensure that the results are not overfitted with the guidance of physics knowledge.

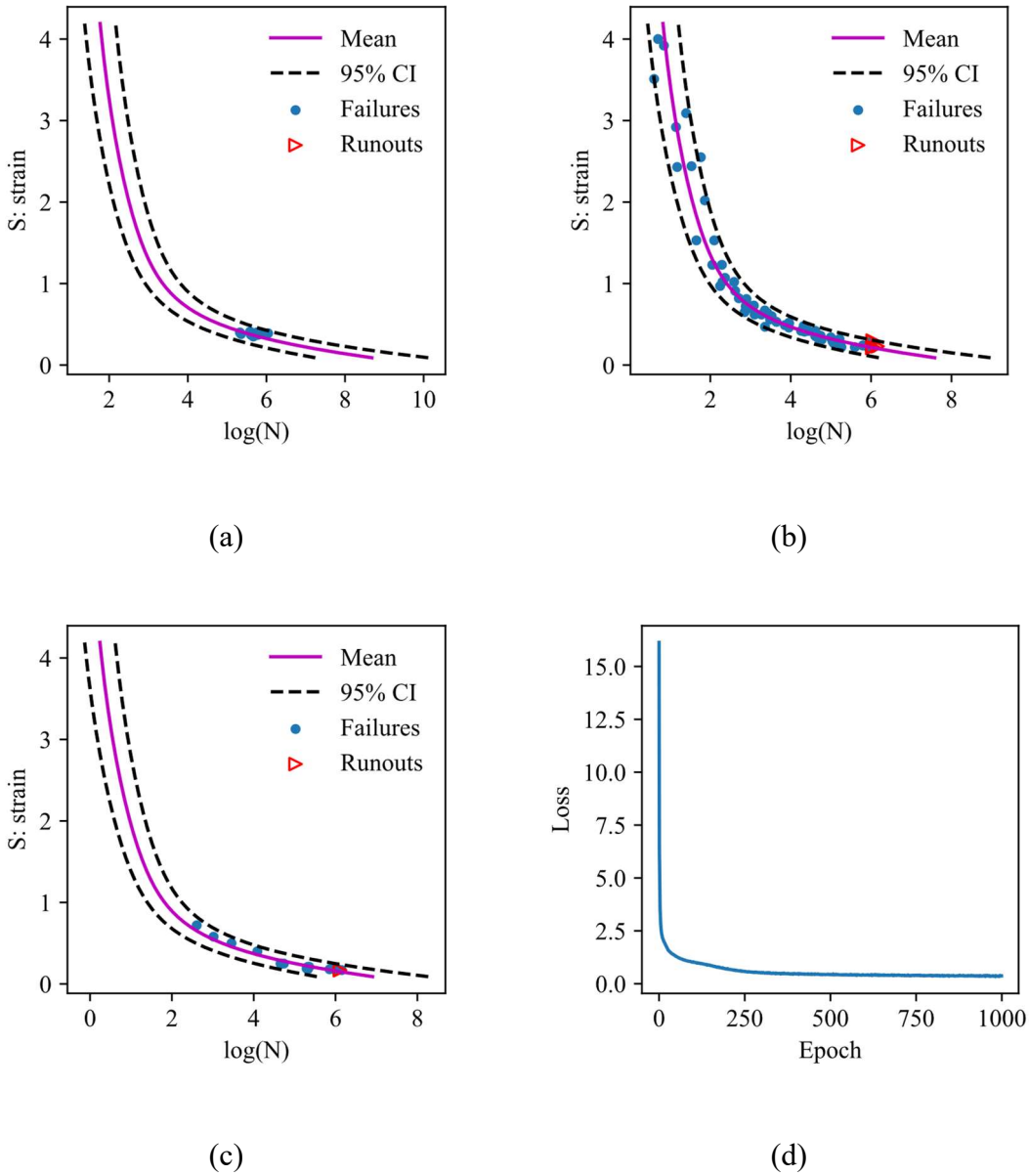


Fig. 3.7 Validation of PPgNN Using Multi-factor Experimental Data with Mean Stress Effect.

3.2.6 Discussions

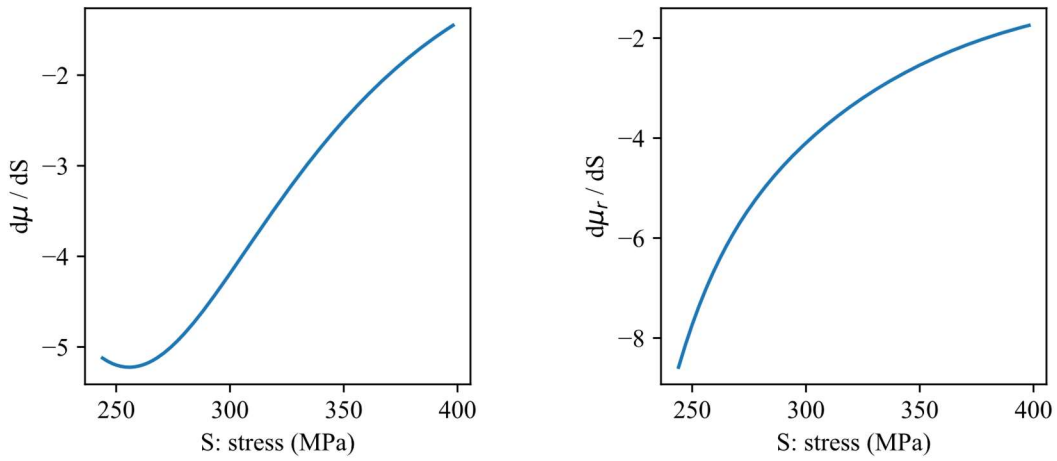
The PPgNN architecture is validated using fatigue testing data for single and multi-factor in Section 3. In his section, we discuss the impacts/benefits of the physics guidance in the neural network for fatigue data analysis by comparing results from the neural network in Fig. 3.2 (without physics guidance) and Fig. 3.4 and Fig. 3.5 (with physics guidance). Also, the effectiveness of the relaxation on the mean prediction by adding another output layer is discussed and the results from the neural network architectures in Fig. 3.3 (without relaxation on mean) and Fig. 3.4 (with relaxation on mean) are compared.

3.2.6.1 Impact of the Physics Guidance in PPgNN

The laminate panel data set is used again for the illustration of the necessities for imposing physics knowledge in the neural network. The neural network architecture in Fig. 3.2 produces results that are learned solely from the training data. Without any physics guidance, the results are shown in Fig. 3.8 (a). The improvements of the proposed PPgNN compared with the neural network without any physics guidance are reflected in the curvature and extrapolation performance below the minimum testing stress. The slope $d\mu$ / dS in Fig. 3.8 (a) does not change in a monotonic manner. Below some stress level (about 260 MPa), $d\mu$ / dS increases as the stress decreases. This is contradictory to the common knowledge as the slope will increase as the stress decreases. If there is a fatigue limit, the *S-N* curve will asymptotically approach infinity for low stress values. The reason for

obtaining the physics inconsistent results is due to the overfitting. As can be seen in Fig. 3.6Fig. 3.6 Validation of PPgNN Using Single Factor Experimental Data for Various Materials.

(a), fatigue life data are distributed at five stress levels. The overfitted mean *S-N* curve can pass through the mean of collected data at each of the five stress levels. Due to the data censoring, the mean fatigue life at the low stress levels calculated from the collected failure data cannot reflect the true distribution mean. Thus, by tracking the sample mean, the overfitted *S-N* curve cannot have the desired monotonic slope decrease as stress level decreases. This issue is solved by imposing this physics knowledge while training the neural network, which is done by solving the optimization problem with bias and weight constraints. The improved results obtained from PPgNN are shown in Fig. 3.8 (b). It can be seen that $d\mu$ / dS decreases monotonically as the stress decreases, and the *S-N* curves tend to be flat when approaching the fatigue limit.

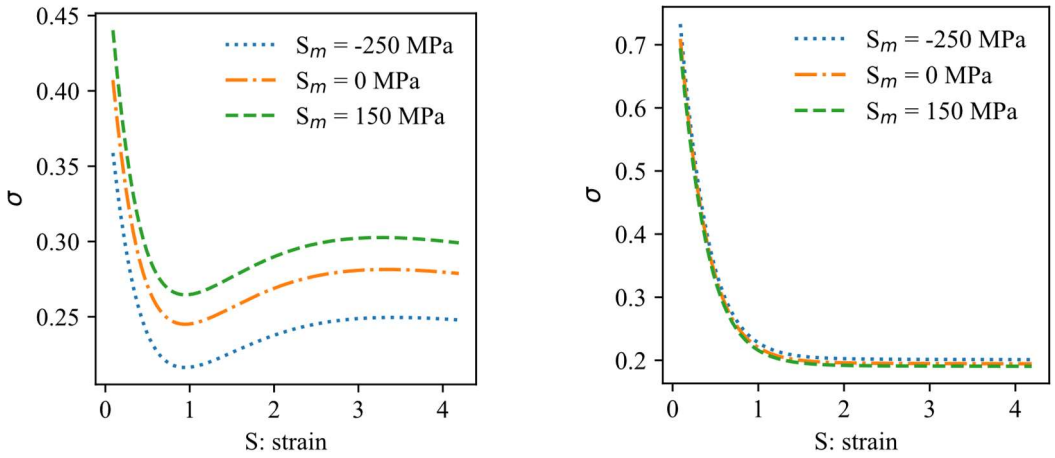


(a) (b)

Fig. 3.8 Comparison of Slope Behavior of (a) Classical NN Without Physics Guidance and (b) Proposed PPgNN with Physics Guidance.

The multi-factor experiment is also refitted using Al 7075-T651 data and the neural network with no physics knowledge. The neural network architecture without physics guidance is similar to that in Fig. 3.2. The input layer contains S: strain, mean stress, log (N), and Index. One hidden layer with 20 neurons which is the same number as that in the architecture in Fig. 3.5. The output layer has μ and σ. The results are shown in Fig. 3.9 (a) for the variance variation. It is seen that the standard deviation for each of the three mean stress cases does not decrease monotonically as the strain level increases. This is also due to the overfitting. Without physics knowledge guidance (i.e., bias and weight constraints), the neural network is learned solely from the data. Since the sample size is not big enough,

the overfitting issue occurs. The prediction with physics guidance is shown in Fig. 3.9 (b) and it is clearly seen that the variance shows a monotonic behavior. It is also interesting to see that the uncertainty in this material is independent of mean stress with the proposed method.



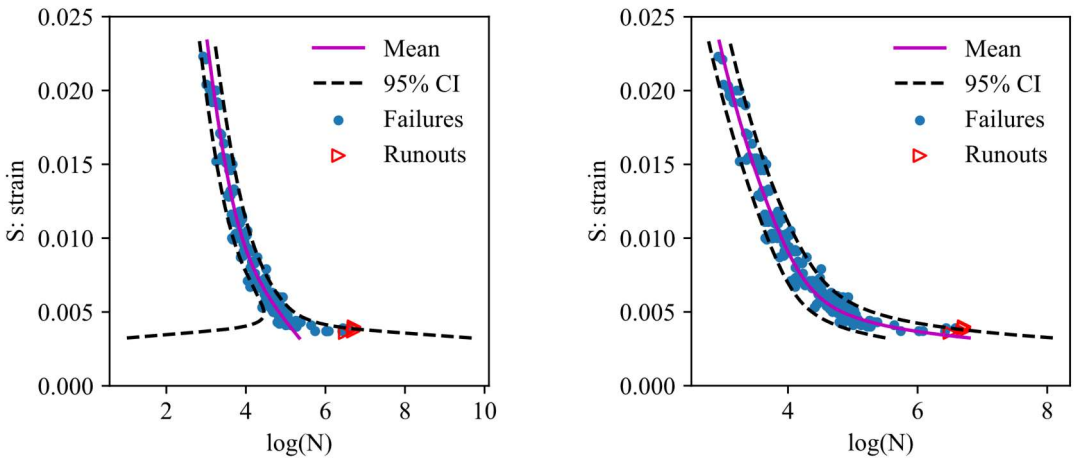
(a) (b)

Fig. 3.9 Comparison of Variance Variation of (a) Classical NN Without Physics Guidance and (b) Proposed PPgNN with Physical Guidance.

3.2.6.2 Impact of the Relaxation Layer in PPgNN

The nickel-base superalloy data set is used to illustrate the effectiveness of the relaxation on the mean of fatigue life by adding another output layer with the neuron μ_r and connecting μ_r with μ and σ at the previous layer. The results using the neural network architecture in Fig. 3.3 without relaxation on μ are plotted in Fig. 3.10 (a). Compared with the results in Fig. 3.10 (b) using the architecture with relaxation on μ, an obvious difference

is reflected in the low strain region. In this region, the fatigue life data tend to asymptotically approach the fatigue limit and the curvature should be close to zero. For Fig. 3.10 (a), the mean *S*-*N* curve can hardly capture this trend of the data. This is due to the over-restrictive constraints for constraint (3.22). With the over-restricted constraints, the second derivative of μ with respect to the strain cannot achieve the value of zero, which leads to that the first derivative of μ with respect to the strain cannot achieve an adequate small negative value. By adding another output layer with the neuron μ_r and connecting μ_r with μ and σ at the previous layer, the imposed over-restricted constraints can be relaxed, and the fitted *S*-*N* curve has more flexibility to follow the trend of the data at the low strain level. The values of $d\mu / dS$ are shown in Fig. 3.10 (c) and (d) for the two approaches. It is clearly seen that the proposed PPgNN with relaxation can successfully predict zero curvature at a very low strain region.



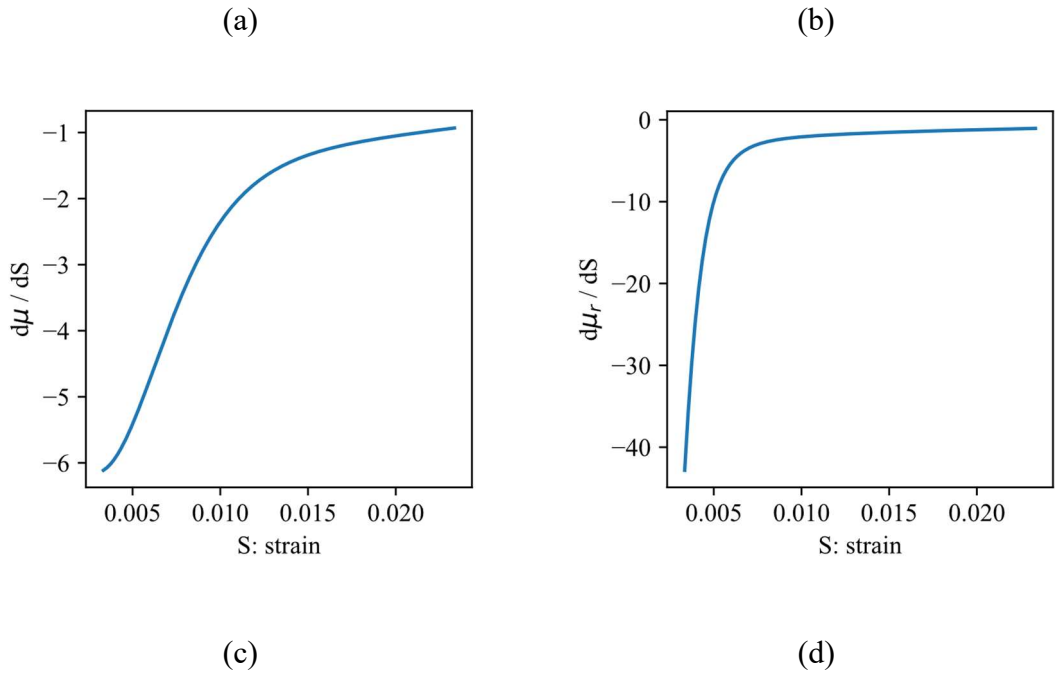


Fig. 3.10 Comparison Prediction Results of (a) Strain-life Prediction Without Relaxation in NN, (b) Strain-life Prediction with Relaxation in NN, (c) $d\mu / dS$ Prediction Without Relaxation in NN, and (d) $d\mu / dS$ Prediction with Relaxation in NN.

3.3 Fatigue Property Prediction of Additively Manufactured Ti-6Al-4V Using Probabilistic Physics-guided Learning

3.3.1 Introduction

Additive Manufacturing (AM) comprises net-shape production technologies that build a solid object from the sequential superposition of layers representing the cross-sections [111, 112]. Compared with conventional manufacturing (for example, casting as well as secondary processing steps such as rolling and forging), AM offers advantages of

freedoms for complex geometries, minimal tooling, rapid delivery times, and low material waste [111, 113, 114]. Selective laser melting (SLM) is one of the commonly used AM processes. In SLM, metallic powders are spread into a thin layer, selectively melted by a finely focused laser, and fused to the previous layer [115]. Ti-6Al-4V (referred to as Ti-64 here afterward) is extensively explored for processing via SLM routes owing to its high strength, low density, biocompatibility, and excellent corrosion resistance [113, 116-118]. In real-life application of Ti-64 such as compressor and turbine blades of aircraft engines, the components are subjected to cyclic loadings which may result in fatigue failure [119]. Therefore, the fatigue characterization of SLM manufactured Ti-64 plays a vital role in ensuring structural safety.

The major factors causing the fatigue failure of AM components by SLM are surface quality, residual stresses, pores, and microstructure [120-122]. Several ways are employed to control AM quality individually or collectively using proper control of process parameters. Regarding the in-process parameters, laser power, scanning velocity, hatch distance, and powder layer thickness are considered as key influencing factors during the SLM processing. Post-process heat treatment is considered essential to reduce residual stresses, close undesired pores and transform the microstructure to improve mechanical properties. Considerable research has been carried out to investigate the influence of these in-process and post-process parameters on fatigue properties of SLM manufactured Ti-64

[123, 124]. In the experiment of Fousová et al. [125], specimens were fabricated by SLM and heat-treated without any further treatment. The fatigue strength was tested to be 240 ± 24 MPa. Wycisk et al. [116] performed high cycle fatigue tests for as-built, polished, and shot-peened samples produced by SLM. As-built samples showed a drastic decrease of fatigue limit due to poor surface quality. Khalid Rafi et al. [126] conducted fatigue testing with specimens fabricated by SLM and heat-treated for stress relief. Fractography observations showed that the fatigue crack initiation occurred deep in the subsurface. Fatemi et al. [127] studied the surface roughness effect by considering both as-built and machined and polished AM specimens. It was found that AM specimens had significantly shorter lives compared to the wrought specimens. Chastand et al. [128] discovered that multiple types of defects contribute to crack initiation for fatigue specimens built by SLM including surface, internal unmelted zones, and small internal defects.

Current studies on fatigue properties of SLM fabricated Ti-64 mainly focus on experimental investigation. There is a lack of reliable modeling between process parameters and fatigue performance, especially considering huge uncertainties in fatigue. With the available testing data from reference literature, a data-driven method is worthwhile to explore for fatigue characterization of Ti-64 produced by SLM. Machine Learning (ML) provides an avenue to gain insight into AM process by learning fundamental knowledge from experimental data and is a valid way to perform complex

regression analysis. Among ML algorithms, the neural network (NN) is the most widely used model due to its flexibility to handle arbitrary nonlinear mapping problems, strong computational power, and sophisticated algorithm architecture [129, 130]. Challenges remain when applying NN to modeling fatigue properties with process parameters due to: (1) A large amount of uncertainties exists in fatigue modeling for AM fabricated components. Fatigue lives show a significant scatter range even at well-controlled testing conditions [88]. Also, various sources of uncertainty are involved in the AM processes from powder bed forming to melting and solidification [131-133]. Thus, a deterministic approach using classical NN is inadequate. (2) Fatigue stress-life (S-N) curves show an increase of variation as stress level decreases [16]. If the NN is solely trained from the data, those physics may be violated due to overfitting. (3) Classical NN requires the data set to be complete, i.e., no missing data for each input variable. However, it is very common that the data sets available from open literature are incomplete. Some of the process parameters are not reported. According to the authors' collection of data, more than half of the data sets contain missing data. Efforts are needed to make use of these incomplete data sets instead of brutally abandoning the valuable information from such data sets. (4) NN is a stochastic algorithm in practice due to the stochastic process for training, random initial weights, etc. Different results are obtained even with the same NN architecture and training

data. The influence of randomness using NN needs to be investigated for reliable AM data analysis.

The proposed study aims to estimate the probabilistic S-N (P-S-N) curves and investigate the effect of process parameters on fatigue properties of SLM manufactured Ti-64 through probabilistic physics-based data-driven modeling. A database is formed by collecting the experimental fatigue data from extensive literature subjected to various in-process and post-process parameters. The classical NN is modified to provide a probabilistic tool for uncertainty quantification. Both mean and variance of the fatigue life can be learned with respect to fatigue parameters (stress amplitude and stress ratio) and process parameters (laser power, scanning velocity, hatch distance, powder layer thickness, and heat treatment). The proper constraints are imposed on the proposed NN for obtaining P-S-N curves with the physics-consistent curvature and nonconstant variance. To make use of the data sets with incomplete inputs, a novel NN architecture is applied for SLM manufactured Ti-64 figure data analysis. The idea is that the results learned using data set with complete inputs are corrected with the information obtained from the data set containing missing data. With a larger data size, the data-driven approach can provide more reliable predictions.

3.3.2 PPgNN for Fatigue of SLM Manufactured Ti-64 with Missing Data

With the PPgNN introduced in the last section, the architecture is extended to be suitable for fatigue property analysis of SLM manufactured Ti-64. The objective is to study the influence of process parameters on fatigue life. The in-process parameters considered are scanning velocity (v), laser power (P), hatch distance (h), and powder layer thickness (t). The post-process parameters considered are heat temperature (HT) and heat time (Ht). The data used for NN training are obtained from reference literature. It is observed that among all the collected data, more than half of the data are incomplete. That is, the input data contain missing data. To fully use all the available data instead of abandoning the incomplete data set, the NN architecture is redesigned to consider the missing data.

The proposed architecture is shown in Fig. 3.11. No constraints are imposed on biases and weights between process parameter inputs and outputs. This is based on the considerations that the physics knowledge between process parameters and fatigue life (mean and standard deviation) is unclear, and this relationship is intended to be learned from the data. The neurons in square shape are called selective neurons. They are designed to enable NN to be trained with an incomplete data set [134]. The composition of the selective neuron is shown below the NN architecture in Fig. 3.11. It comprises a hidden neuron with tanh activation function, an input neuron r_k indicating if an input is available or missing and a multiplier. Each process parameter input neuron is connected to a selective

neuron. No selective neuron is connected to fatigue parameters (S and R) since these data are usually available from reported data. The mathematical expression for the selective neuron is

$$z_k^{(s)} = b_k^{(1)} + w_{k1}^{(1)} x_k , \tag{3.31}$$

$$a_k^{(s)} = r_k g_k^{(s)} \left(z_k^{(s)} \right) \quad k = 1, 2, ..., p_s , \tag{3.32}$$

where the superscript (s) represents the layer which contains selective neurons (Layer s), and p_s is the total number of selective neurons. If the value of x_k is available (r_k = 1), the selective neuron will actively participate in the computation of the network outputs. If x_k is missing, r_k = 0, which behaves as the input x_k does not exist. When some input is missing, the bias and weight of the selective neuron connected to this input will not be updated. The architecture with selective neurons can be viewed as a collective of multiple models. Each model contains a certain number of inputs whose values are available (not missing). They are bonded to each other and share information [134].

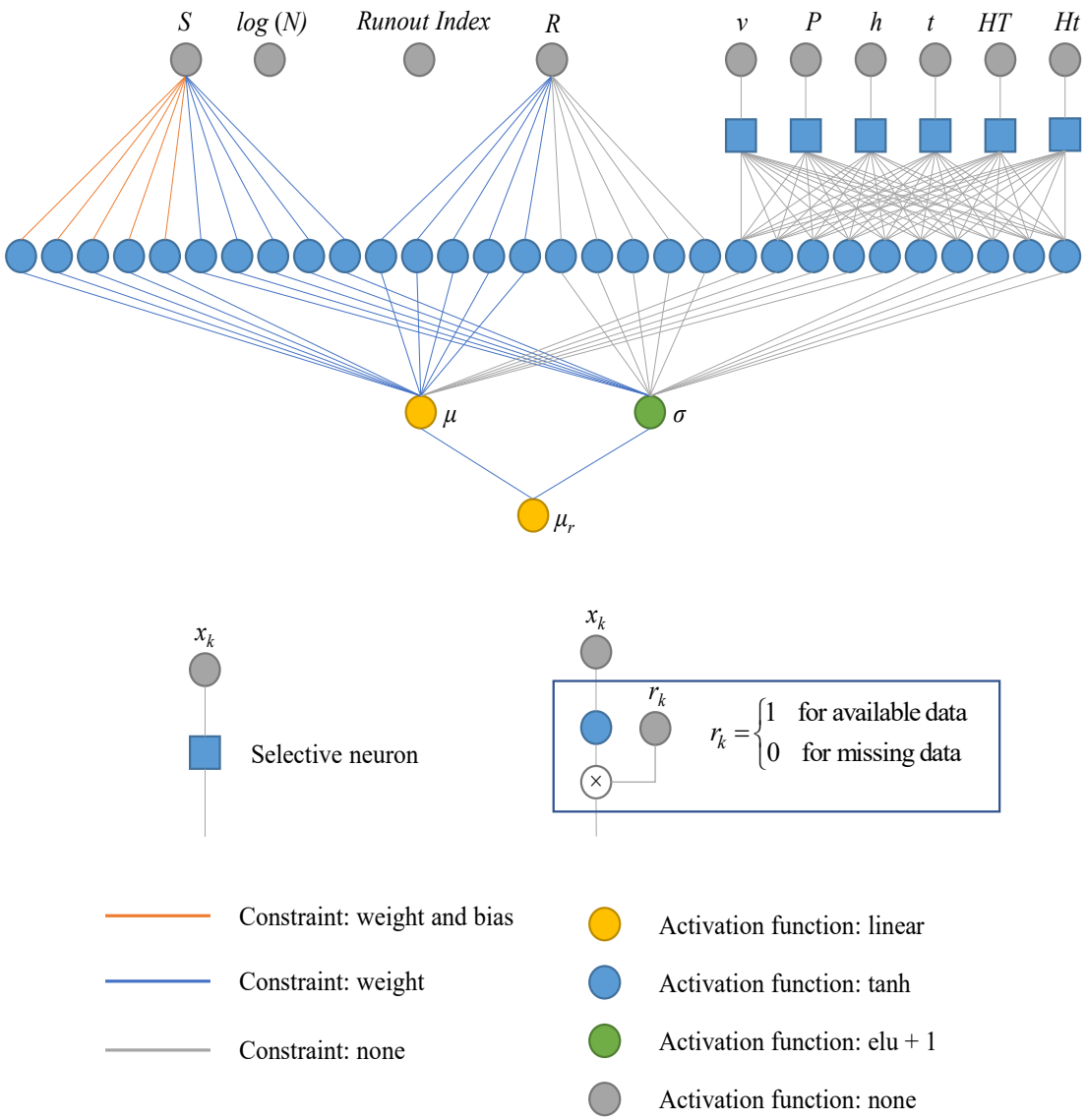


Fig. 3.11 Probabilistic Physics-guided Neural Network for Fatigue Analysis of SLM Manufactured Ti-64 Considering Missing Data.

The method used is an algorithm which has built-in support for missing data. Another way to deal with missing data is preprocessing such as data imputation. Data imputation is a technique to replace the missing data with data imputed to form a complete

data set according to the correlation between variables. Data imputation depends on a missing mechanism such as missing at random and not missing at random [9]. The algorithm used in this work does not require the knowledge of missing mechanisms since only the observed data are used for computation [134].

3.3.3 Predictive Performance

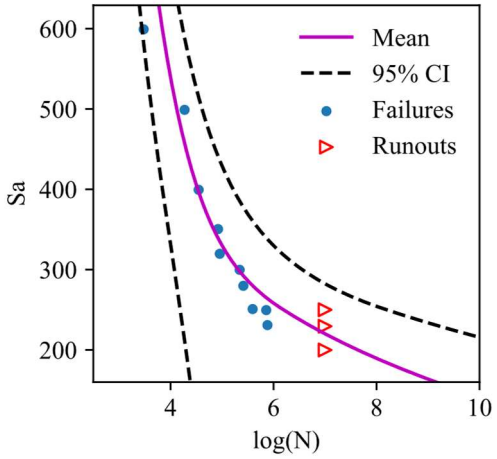
The experimental data are collected for fatigue property analysis of SLM fabricated Ti-64 from the literature. The specimens are manufactured with different in-process parameters (scanning velocity (v), laser power (P), hatch distance (h), and powder layer thickness (t)). After fabrication, the heat treatment is performed with different post-process parameters (heat temperature (HT) and heat time (Ht)). All the specimens are in as-built and annealed conditions without machining, polishing, or hot isostatic pressing treatment. The summary of the collected data is shown in Table 3.3. Each of the data sets has a set of specific process parameters from different references. The "NA" represents the data that are not reported (missing data) from the references. A total of 15 data sets are collected from 11 references. The first 6 data sets are comprised of complete data. The last 9 data sets contain missing data for inputs.

Table 3.3 Summary of Collected Fatigue Data for SLM Manufactured Ti-64 in As-built and Annealed Condition.

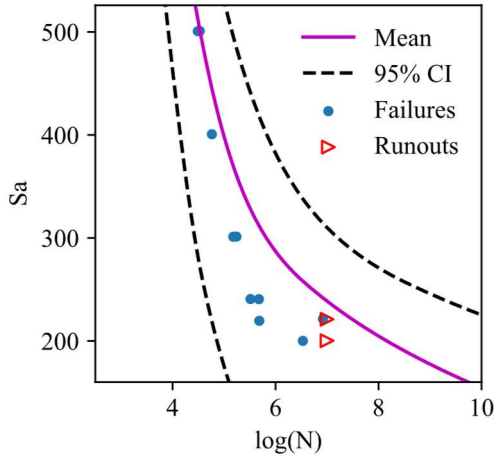
No.	Speed (mm/s)	Power (W)	Hatch space (μm)	Layer thickness (μm)	Heat temperature (°C)	Heat time (h)	Stress ratio	No. of failure	No. of runout	Ref.
1	1250	200	80	30	820	1.5	-1	10	3	[125]
2	1250	170	100	30	650	3	0.1	10	2	[116]
3	1200	280	140	30	704	1	-1	13	3	[135]
4	1250	170	100	30	650	4	0.1	8	1	[126]
5	1000	400	160	50	700	1	-1	5	1	[127]
6	1200	280	140	30	704	1	-1	13	2	[136]
7	NA	200	NA	30	800	4	0.1	9	2	[137]
8	NA	400	NA	60	800	4	0.1	11	2	[137]
9	NA	400	50	60	740	2	-1	17	5	[138]
10	NA	400	50	60	1200	2	-1	6	1	[138]
11	NA	400	50	60	900	2	-1	7	5	[138]
12	NA	NA	NA	NA	670	5	-3	25	7	[111]
13	NA	NA	NA	NA	920	2	-3	10	2	[111]
14	NA	200	NA	30	650	3	0.1	19	2	[139]
15	1000	350	NA	60	800	5	-1	32	2	[140]

With the data in Table 3.3 Summary of Collected Fatigue Data for SLM Manufactured Ti-64 in As-built and Annealed Condition., the PPgNN model is validated by evaluating the predicted probabilistic S-N curves of SLM manufactured Ti-64. Cross-

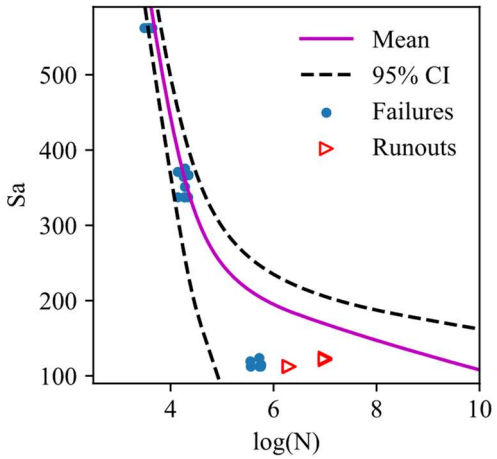
validation is conducted. The strategy is that for each validation, one of the 6 complete data sets is used for testing, and the remaining 5 complete data set as well as the 9 incomplete data sets are used for training. Thus, it is 6-fold cross-validation. Only complete data sets are used for testing, and the incomplete data sets are always used for training. This is because it is not feasible to obtain the predictions with missing data in model inputs. However, with the proposed model, the incomplete data can be used in the training process.

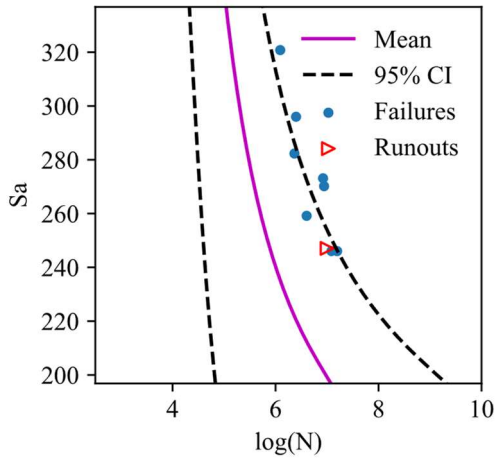


(a) Data set 1.

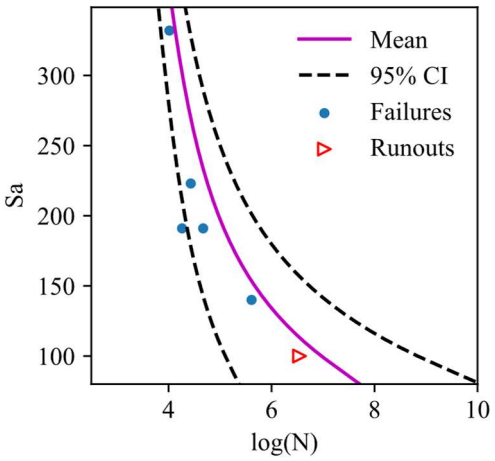


(b) Data set 2.

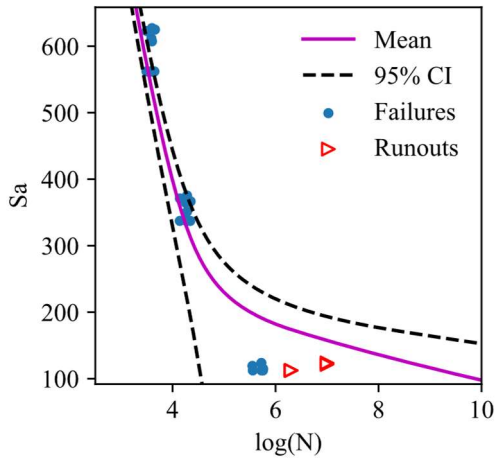




(c) Data set 3.



(d) Data set 4.



(e) Data set 5.

(f) Data set 6.

Fig. 3.12 Predicted P-S-N Curves.

The predictive results are shown in Fig. 3.12. The predicted P-S-N curves as well as the experimental data in Fig. 3.12 (a) – (f) correspond to the data set No. 1 – 6. The x-axis and y-axis represent log fatigue life and stress amplitude, respectively. The mean S-N curves are shown in solid lines and the 95% confidence intervals are shown in dash lines. The experimental data in dots and triangles are the failure and runout data, respectively. It can be seen that with the fatigue physics knowledge imposed in the training of NN, the results obtained show desired features for variance and slope change of P-S-N. The predicted P-S-N curves can reasonably quantify both mean values and distribution of fatigue lives for most validation cases. Almost all the experimental data are located within the predicted confidence bounds.

The proposed PPgNN model is capable of taking incomplete data into account for training. By considering the incomplete data, the data size is larger and more information can be learned for NN from the data compared with using only the complete data. In this section, the predictive performance of the PPgNN model is compared between results obtained by training the model using all the data (both complete and incomplete data) and only the complete data.

The architecture used for all data is as shown in Fig. 3.11 with the selective neurons for missing data. Since no missing data exit in the complete data set, the model for the complete data has the architecture shown in Fig. 3.11 without the selective neurons. The logarithmic scoring rule [8, 88] is used as a metric for comparing the predictive performance of both models. The expression of the log score is the same as the loss function shown in Eq. (4). The loss function is used in the training process, while the log score is obtained using the testing data. A smaller log score shows a better prediction. Due to the randomness of the neural network, the log score is calculated 50 times, each of which corresponds to a trained model with random initial biases and weights. The log scores for models using all data (with missing data) and only complete data (without missing data) are presented in Fig. 3.13. Same as the strategy in Section 3.2.1 and 3.2.2, cross-validation is conducted. The x-axis indicates the complete data set used for testing. In each testing data set, a log score can be calculated for every experimental datum (both failure and

runout). The mean value of all the log scores in one testing data set is plotted to show the average predictive performance. The "Overall" on the x-axis indicates the overall predictive performance for the model trained by all data or only the complete data. The mean log scores from all the cross-validations are plotted together for evaluating the overall performance. For each testing case, the box plots on the left and right are the results for using all data and complete data, respectively. The thick line in the middle part shows the median value. The lower and upper bounds of the rectangular are the 25^{th} and 75^{th} percentile, respectively. The lower and upper bound of the entire boxplot are the minimum and maximum excluding the outliers, respectively.

For testing data set No. 1 and 2, the median mean log score is lower if training the model using all data. For test data sets No. 3, 4, 5, and 6, both modes have comparable median mean log scores. The range between minimum and maximum may be shorter or longer for the model using all data compared with the model using complete data depending on the testing data set. Overall, the median mean log score is smaller and the range between minimum and maximum is shorter for the model trained by all data. This shows the improvement for prediction if both complete data and incomplete data are used as training data with the PPgNN model.

The histogram distributions of all the mean log scores are plotted in Fig. 3.14 corresponding to the overall performance in Fig. 3.13. A narrower distribution can be

observed for the model trained using all data (with missing data). This shows that by considering the incomplete data set for training, the model is more likely to produce good predictions since more information is provided compared with only using the complete data (without missing data). Thus, it is a good practice for fatigue analysis of SLM manufactured Ti-64 to fully use the incomplete data set.

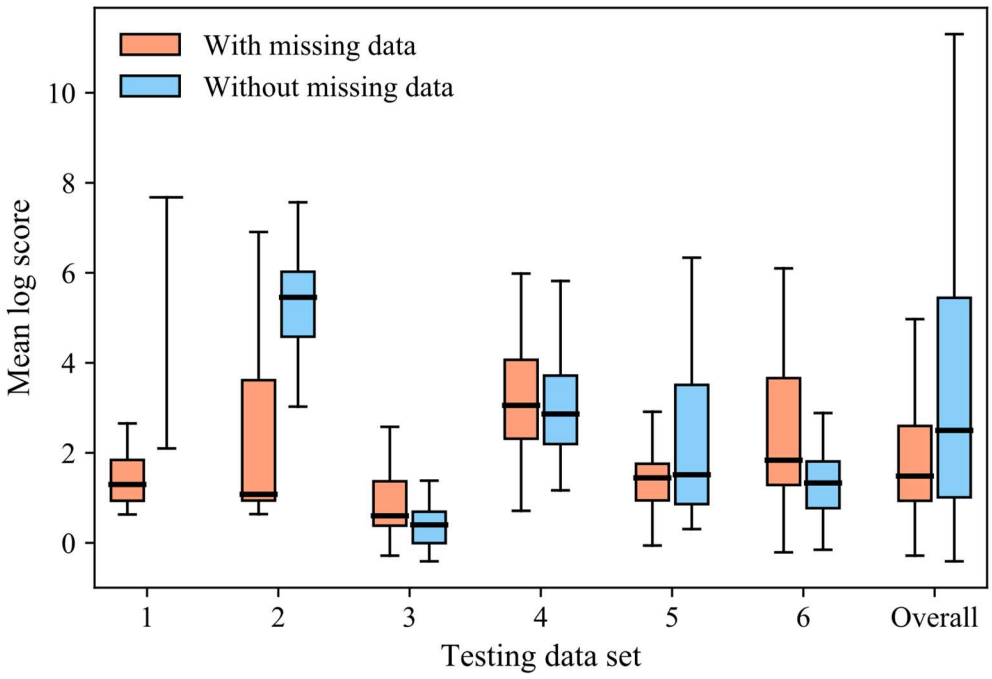


Fig. 3.13 Predictive Performance Comparison Between Model Trained Using All data and Model Trained Using Only Complete Data.

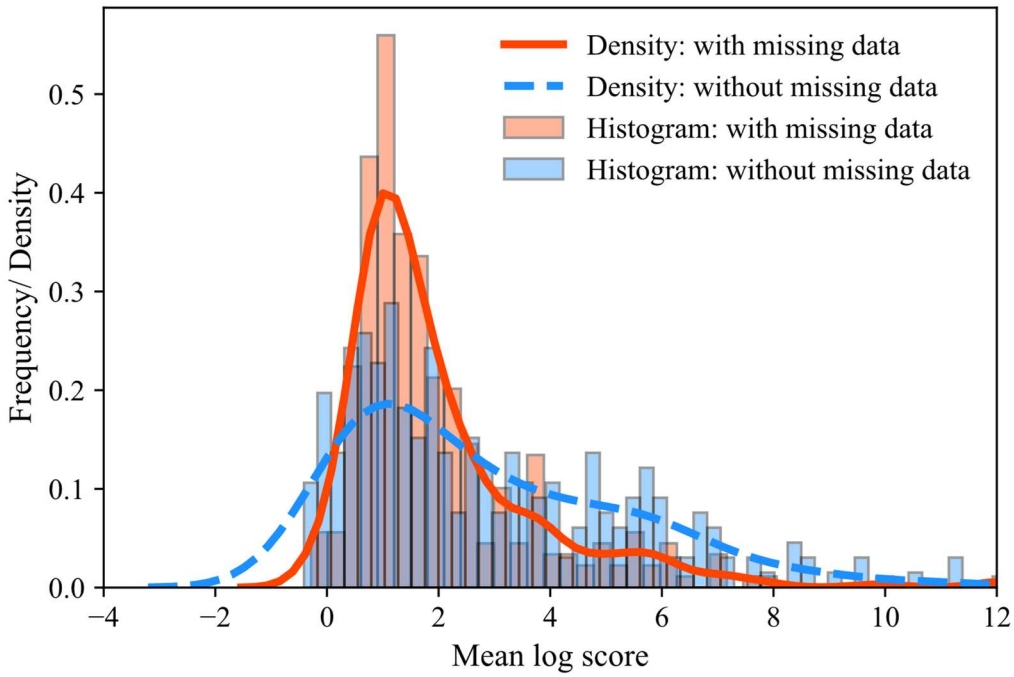


Fig. 3.14 Histogram Distribution of All Mean Log Scores.

3.4 Physics-guided Mixture Density Networks for Uncertainty Quantification

3.4.1 Introduction

Complex systems require simulation models to represent the underlying scientific phenomenon flexibly and accurately [141]. Uncertainty Quantification (UQ) in the predicted response of a system model is essential to establish confidence in the representation of actual system behavior [142-144]. The choice of data distribution is crucial for the UQ. For example, in survival data analysis, the probability of failure estimate depends heavily on the tail region of the lifetime distribution [106]. However, with little prior knowledge and/or sparse data of a complex system, it is not an easy task to determine

an appropriate distribution. Extensive studies have been proposed using classical statistical analysis, Bayesian statistics, and neural networks for UQ of engineering systems [67, 85, 106, 145, 146].

The current study focuses on one type of model, i.e., Mixture Density Networks (MDN), due to their flexibility for UQ. The MDN is a combined structure of a feedforward neural network and a mixture model [147]. It can provide a flexible mean estimation due to the neural network component. Also, the uncertainty quantification can be made without any assumption on the distributions due to the mixture distribution component. Hence, it can deal with complex patterns and arbitrary distributions found in real-world data. Recent applications of MDNs are summarized as follows. Davis et al. [148] introduced the concept of MDNs in the emulation of epidemiological models. Lovell et al. [149] explored the use of the MDN to reproduce fission yields and their uncertainties and showed that the MDN provides a reliable way to estimate uncertainties for nuclear data. Yang et al. [150] used MDNs combined with generative adversarial networks for inverse modeling in materials microstructural design. Men et al. [151] applied an ensemble approach based on MDNs to the point and probabilistic forecasting of wind speed and wind turbine power. Zhao et al. [152] applied MDNs with a deep bidirectional long short-term memory model to predict basketball trajectories based on real data and generate new trajectory samples.

The flexibility of MDN for any complicated functional and distributional form, however, does increase the required data size to train the model. This may introduce difficulties when using MDNs for computationally expensive models [148]. Extensive physics knowledge developed for years or centuries can be used to mitigate this issue, which has not been utilized in the modern machine learning practice [153]. The existing physics knowledge may be either classical mathematical models or empirical relationships developed within a discipline [154]. Serving as a regularization role, the prior information constrains the space of admissible solutions for a machine learning model to explore. Thus, it guides the learning algorithm towards the right solution with a small data size [72]. The concept of physics-guided neural networks was introduced in [153] for solving forward and inverse problems involving partial differential equations. The studies of [154, 155] extend the deterministic physics-guided neural networks in [153] through dropout and Bayesian methods to quantify the uncertainty. Those methods may suffer from undefined/pathological behavior [154, 156] or an inappropriate assumption of prior distributions for model parameters [155]. Also, it is a challenging problem if the data distribution is non-Gaussian or the data includes incomplete data set (e.g., censored data).

An algorithm is proposed, named Physics-guided Mixture Density Networks (PgMDN), to address the above-mentioned difficulties. It has the following novelties. First, this model can quantify uncertainties and no assumption of the data distribution is needed. The

distribution type is also flexible to change at different inputs. That allows reliable uncertainty quantification in a complex system. Second, the physics knowledge of the derivative information can be incorporated into the neural network. With the guidance of prior knowledge, the required number of training samples can be reduced, and desired physics-consistent results can be obtained. Next, a dynamic penalty function method is integrated with the PgMDN for the constrained optimization, which automatically balances the data and physics knowledge in the training process. Finally, this model does not impose any limitations on the neural network architecture (e.g., number of hidden layers, activation functions). A wide range of physics constraints (e.g., value constraints or derivative constraints, equality or inequality constraints) can be properly designed to be incorporated into the neural network.

3.4.2 Physics-guided Mixture Density Networks

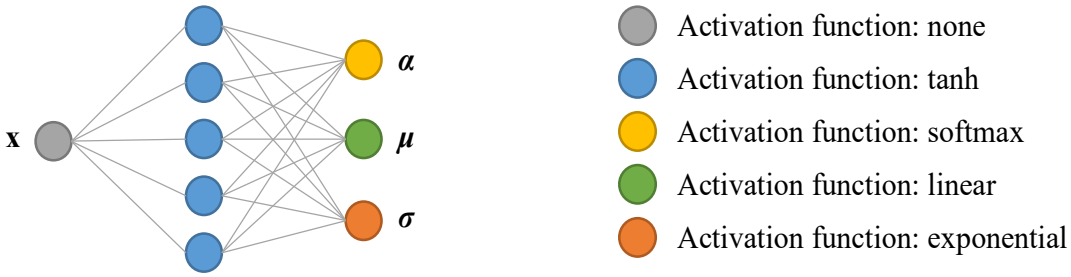


Fig. 3.15 An Illustration of Mixture Density Network Architecture.

The architecture of the Mixture Density Network used in this work is illustrated in Fig. 3.15. Three layers are included in this network: the input layer, one hidden layer, and

the output layer. Unlike the regular neural network that only learns the mean values of the outputs from the collected distributed data, there are three output neurons in the output layer: mixing coefficients, mean values, and standard deviations. Instead of the conventional least-square loss, the loss function is the negative logarithm of the probability density of the target data given the inputs, i.e.,

$$L = -\sum_{i=1}^{n} \log p\left(\mathbf{y} \mid \mathbf{x}\right), \tag{3.33}$$

where n is the number of training data. x and y are a set of input and output variables, respectively. p(y | x) denotes the conditional probability density of y given a particular x, which is a mixture distribution of the form

$$p\left(\mathbf{y} \mid \mathbf{x}\right) = \sum_{j=1}^{m} \alpha_j\left(\mathbf{x}\right) \phi_j\left(\mathbf{y} \mid \mathbf{x}\right), \tag{3.34}$$

where m is the number of components in the mixture distribution. α_j(x) is the mixing coefficient which is the probability that the target vector y is generated from the jth component of the mixture. ϕ_j(y | x) is the kernel distribution. Various types of kernel functions can be used. In this work, the kernel functions are chosen to be Gaussian with the form of

$$\phi_j\left(\mathbf{y} \mid \mathbf{x}\right) = \frac{1}{\left(2\pi\right)^{c/2} \sigma_j\left(\mathbf{x}\right)^c} \exp\left[-\frac{\| \mathbf{y} - \boldsymbol{\mu}_j\left(\mathbf{x}\right) \|^2}{2\sigma_j^2\left(\mathbf{x}\right)}\right], \tag{3.35}$$

where c is the number of outputs. μ_j (x) is the mean vector of the jth kernel with c components. σ_j is the standard deviation shared among all the components of the output vector in the jth kernel.

Different colors of neurons in Fig. 3.15 indicate specific activation functions. For the input layer, no activation functions are used by the definition of the neural network. The hyperbolic tangent activation function (named tanh) is used for the neurons in the hidden layer. The expression of the tanh activation function is

$$g(z)=\frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}, \tag{3.36}$$

where z is the output of the corresponding neuron. Other appropriate activation functions can also be applied to the hidden layer. The mean (μ) in the output layer has the linear (i.e., identity) activation function which is usually used for outputs in the regression problem,

$$g(z)=z. \tag{3.37}$$

Since the sum of the mixing coefficients α_j(x) should equal unity, i.e.,

$$\sum_{j=1}^{m}\alpha_j(\mathbf{x})=1, \tag{3.38}$$

the softmax activation function is used for α in the output layer with the form of

$$g(z)=\frac{e^{z}}{\sum_{j=1}^{m}e^{z}}. \tag{3.39}$$

Since the standard deviation must be non-negative, the exponential activation function is used for σ and is expressed as

$$g(z) = e^{z}. \tag{3.40}$$

The total number of outputs is $m \times c$ mean values, m standard deviations, and m mixing coefficients. The distribution parameters (i.e., the outputs of the network: mixing coefficients, mean values, and standard deviations) are functions of the input x, which provide a general tool to model arbitrary distributions for different inputs. After obtaining the distribution parameters, the mean, variance, or any quantile of the mixture distribution at any input can be computed. Thus, uncertainty quantification can be performed with this mixture distribution.

To incorporate the physics knowledge, training a neural network is formulated as a constrained optimization problem:

$$\begin{aligned} &\text{Min } L(\mathbf{x}, \mathbf{y}; \theta) \\ &\text{subject to} \\ &g_p(\mathbf{x}; \theta) \le 0 \quad p = 1, ..., P \\ &h_q(\mathbf{x}; \theta) = 0 \quad q = 1, ..., Q \end{aligned} \tag{3.41}$$

where x and y are input and output variables, respectively. θ is neural network parameters. The objective function L(x, y; θ) is the loss function shown in Eq. (3.33). The physics knowledge is described by the constraints. There are P inequality and Q equality constraints. In this work, the physics constraints are first or higher-order derivatives of

outputs with respect to input variables. In this way, the physics knowledge can be partial differential equations or empirical relationships between outputs and inputs (e.g., the trends or curvatures of the fitted curves).

To train the physics-guided neural network (i.e., to solve the constrained optimization problem in (3.41), two problems remain. The first problem is how to differentiate neural network outputs with respect to the inputs for obtaining the expressions of the constraints. The second one is how to solve the above constrained optimization problem. Those problems are discussed in the following two subsections, respectively.

The physics knowledge is expressed as partial derivatives in the constraints in the optimization problem stated in (3.41). To calculate the derivatives of the neural network model with respect to its inputs, one can manually work out the derivatives and code them. That method was used in [83, 84, 157], which is shown to be beneficial for a specific problem using a simple NN architecture. However, the disadvantage is that the activation functions need to be selected carefully to simplify the calculations. Also, any change to the NN architecture (e.g., activation functions or the number of layers) requires a re-derivation of the analytical derivatives. Another method is automatic differentiation (AD). A specific case of AD in NNs is the backpropagation algorithm. It applies the reverse mode AD to the objective function (loss function) with respect to its set of weights and biases and computes the partial derivatives for the weight and bias update [158]. The same algorithm

is used to differentiate the NN with respect to its inputs instead of weights and biases. In this case, the objective function is expressed as

$$f(\mathbf{y}) = NN(\mathbf{x}), \tag{3.42}$$

where *NN* is the NN model as a function of its inputs x. f (y) is a function of the NN outputs. It can be any single output when using NNs for a deterministic analysis. For the probabilistic analysis using MDN, f (y) can be the mean or variance of the mixture distribution, which is a function of all the outputs α, μ, and σ from the MDN.

The penalty function method is used to solve the optimization problem in Eq. (3.41). Penalty function methods transform the constrained optimization problem to an unconstrained one by alternative formulations [159]. The backpropagation algorithm can still be applied without the need for adaption using this method [159]. The unconstrained optimization problem is expressed as

$$\phi = L(\mathbf{x}, \mathbf{y}; \theta) + \sum_{p=1}^{P} r_p G_p \left[g_p \left(\mathbf{x}; \theta \right) \right] + \sum_{q=1}^{Q} c_q H_q \left[h_q \left(\mathbf{x}; \theta \right) \right], \tag{3.43}$$

where ϕ in the new objective function. G_p and H_q are the functions of g_p(x; θ) and h_q(x; θ), respectively. r_p and c_q are positive numbers known as penalty parameters. The first term of Eq. (3.43) is the objective function in Eq. (3.41). It represents how well a fitted curve can describe the data (data fitting loss). The second and third terms, called penalty terms,

correspond to the inequality and equality constraints (see Eq. (3.41)), respectively. They serve as regularizations due to the physics guidance (physics constraint loss).

The penalty function formulation can be divided into interior and exterior methods. For the interior method, the unconstrained solution is sought from the feasible region. On the contrary, the exterior method minima are infeasible and converge to the solution from the outside. The main drawback of the interior method is that a feasible starting point is needed. However, with many parameters in the NN, a feasible starting point is not readily available. This issue can be circumvented in the exterior method. General formulas of G_p and H_q for the exterior penalty method are [160]

$$G_p\left[g_p\left(\mathbf{x};\theta\right)\right]=\max\left[0,g_p\left(\mathbf{x};\theta\right)\right]^{\beta}, \tag{3.44}$$

and

$$H_q\left[h_q\left(\mathbf{x};\theta\right)\right]=\left|h_q\left(\mathbf{x};\theta\right)\right|^{\gamma}, \tag{3.45}$$

where β and γ are constants (commonly chosen as 1 or 2 [159, 160]). In this work, 2 is chosen for both exponents. If the constraints are satisfied, i.e., g_p (x; θ) $\leqslant$0 and h_q (x; θ) = 0, the penalty terms will be zero. Otherwise, a large number will be added to ϕ so that the solution is forced to move towards the feasible region.

The penalty parameters r_p and c_q control the severeness of the penalty. With a large penalty, the optimization algorithm converges to a feasible solution quickly although it is far from optimal. A small penalty may result in an unfeasible solution. In [153], the loss

function is constructed as a simple summation of the data loss and physics loss. The penalty parameter is set to be one and is fixed while training the NN. This approach is called the static penalty function method. There are several drawbacks for this approach. First, the constant value of one may not be appropriate for the trade-off between data fitting and the physics penalty for different problems. Second, with the same importance attached to data fitting loss and physics constraint loss, both types of loss are minimized simultaneously. The backpropagation algorithm tries to find a solution which minimizes the overall losses. As a result, the solution may lie in the infeasible regions and the physics constraints are violated. To alleviate the above issues, a dynamic penalty function method is proposed for training the physics-guided NN. A summary of the algorithm is shown in Algorithm 3.1. More detailed descriptions are given below.

Algorithm 3.1	Physics-guided Neural Network optimization using dynamic penalty function method.

1. Train the NN with loss function Eq. (3.43). Early stop when the data fitting loss converges.
2. Check whether all the constraints are satisfied. If the physics constraint loss is equal to zero, NN is well trained and hence terminate the procedure. Otherwise, go to Step 3.

3. Increase the penalty parameters, decrease the learning rate, and set the initial weights and biases as the results from the previous iteration.
4. Train the NN with the loss function Eq. (3.43). Early stop when physics constraint loss converges. Go to Step 2.

Step 1. The NN is trained with an arbitrary set of initial model parameters (weights and biases). The loss function is the penalty function in Eq. (3.43). In this work, the penalty parameters are 1 for this step. The data fitting loss is monitored. The NN training stops when the data fitting loss converges. The purpose of this step is to fit the data under the physics constraints with relatively small penalty parameters. The physics constraint loss at the end of this step may not be zero. That problem is handled through the following steps.

Step 2. This step determines whether further iterations of the training are needed. The entire procedure terminates when the physics constraints are satisfied. If not, the NN is trained again with an adjusted set of parameters.

Step 3. In this step, a new set of parameters is prepared for the next iteration of training in Step 4. The focus is on the physics loss. The updated values of the penalty parameters r_p and c_q are greater than those in the previous iteration. The initial weights and biases are assigned to the values at the end of the previous iteration. The learning rate for the next iteration is also decreased. The decreased learning rate is to prevent the updated NN weights and biases from moving too much from the previously trained results. This is

because the results from Step 1 can fit the data reasonably well with some violations of the physics constraints. This step is to adjust the NN to satisfy the physics constraints incrementally by increasing the penalty parameters while keeping the data fitting loss not changing dramatically. With a large learning rate, the NN cannot fit the data well when the NN pays much attention to the physics constraint loss. In this work, the parameters are increased or decreased by being multiplied or divided by a factor greater than 1, respectively.

Step 4. The next iteration of the NN training starts with the parameters from Step 3. The physics constraint loss is monitored. The NN training stops when the physics constraint loss converges. Then go to Step 2 for continued calculation.

3.4.3 A Numerical Example

A numerical example is presented to illustrate the performance of data fitting using the PgMDN. The simulated data are generated according to

$$y = 10\sin(2\pi x) + \varepsilon, \quad \varepsilon \sim Triangular(-3, 8, 0), \tag{3.46}$$

where the error ε is distributed according to the triangular distribution with the lower limit, upper limit, and mode of -3, 8, and 0, respectively. The triangular distribution is selected since it is a strongly non-Gaussian distribution and cannot be approximated properly by a single Gaussian distribution. The input x belongs to range [0, 0.5]. 200 and 30 data points are randomly generated for range [0, 0.3) and [0.3, 0.5], respectively. This non-uniform

generation of data is to mimic some engineering applications, where some certain range of data is easier/cheaper to obtain. Fig. 3.16 shows the simulated data points ("+" symbols) and ground truth mean (black solid curves) and 5% and 95% quantile curves (black dashed curves). The ground truth curves are calculated by sampling 100 million data from the true triangular distribution.

Fig. 3.16 (a) shows the fitted results using the proposed PgMDN model with the dynamic penalty function method for the mean and 5% and 95% quantile curves (red curves). The physics constraint is that the variance is the same everywhere. That is, the first derivative of the variance with respect to the input is zero. This corresponds to the equality constraint in Eq. (3.41) and Eq. (3.43). There are no inequality constraints in this example. The MDN architecture shown in Fig. 3.15 is used. Adam optimization is employed. 1 hidden layer with 10 neurons is used. The activation functions are the same as those shown in Fig. 3.15. The penalty parameter c in Eq. (3.43) is 1 initially and increased by a factor of 5 for each of the following iterations, i.e., $c_{k+1} = 5\ c_k$, where k is the number of the iteration. The initial learning rate lr_0 is 0.01 and $lr_{k+1} = lr_k / 2$ for the subsequent iterations. There are 3 components in the mixture distribution. From Fig. 3.16 (a), it can be seen that the estimated 5% and 95% quantiles are close to the ground truth. It is observed that the variance is unchanged at any input. That is achieved due to the variance constraint. The results in Fig. 3.16 (a) are obtained using the dynamic penalty function

method. The static penalty function method is also conducted for comparison. The results are shown in Fig. 3.16 (b). The penalty parameter c is fixed to be 1, and the neural network is trained only once. Other than that, the model parameters are the same as those used for Fig. 3.16 (a). It should be noted that this strategy is used in [153]. In Fig. 3.16 (b), the purple solid and dashed lines are the fitted results for the mean, and 5% and 95% quantiles, respectively. Compared with the ground truths, reasonable results can be obtained with the static penalty function method. However, it is observed that the variance is not constant in the simulation domain. For example, the distances between the 5% and 95% quantiles in the lower right and upper middle windows are different. This is because the data fitting loss and physics constraint loss are minimized simultaneously in the static penalty function method. When the NN converges, the physics constraint loss is minimized but is not zero. Thus, the prescribed physics constraints are not rigorously satisfied (i.e., variance is not constant). This comparison clearly shows that the dynamic penalty function method can ensure that the physics constraints are better enforced. Fig. 3.16 (c) shows the results obtained using the MDN without any constraint. Due to the data sparsity in the range [0.3, 0.5], the fitted 95% quantile curve deviates from the ground truth. This observation shows the benefits of physics knowledge when the training sample is not sufficient, which is known as one of the general advantages of physics-guided learning [9].

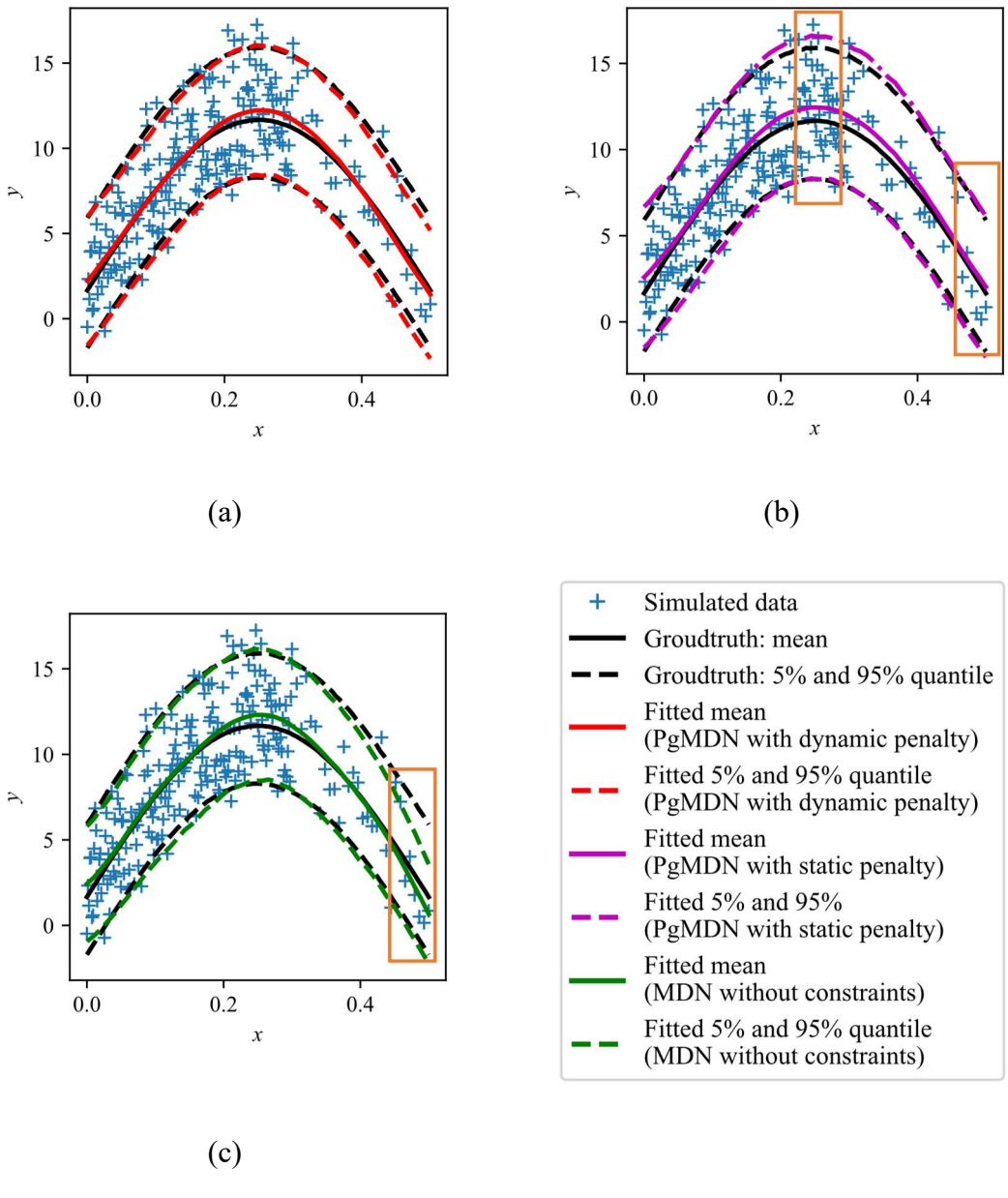


Fig. 3.16 Fitted Curves for the Mean and 5% and 95% Quantiles. (a) PgMDN with Dynamic Penalty. (b) PgMDN with Static Penalty. (c) MDN Without Constraints.

3.4.4 Probabilistic Fatigue Stress-life Curve Estimations

The fatigue failure analysis is a specific case of the survival analysis [60]. The relationship between probabilistic fatigue lives (N) of materials and applied stresses (S) is required for the safe-life design processes considering reliability [9, 16, 44, 83-88, 161, 162]. Those relationships are usually presented in the form of probabilistic S-N (P-S-N) curves [89]. The logarithm of the fatigue life at a stress level is usually modeled using a Gaussian distribution or weakly non-Gaussian distributions (e.g., Weibull) for the finite life regime (high stress levels) [86]. However, when the stress level is low, an infinite or an extremely large number of fatigue cycles can be sustained. The data in that regime are sometimes right censored (known as runouts in fatigue). In that regime, the fatigue life cannot be properly described by a Gaussian distribution. Thus, mixture distribution can be used. Another example is that bimodal distribution is observed in fatigue stress-life analysis for certain types of materials due to different mechanisms of fatigue failures [46]. The mixture distribution is more appropriate under these conditions.

Numerous experiments show that the fatigue P-S-N curves have some common characteristics for different materials. For example, the variability of the fatigue life increases as the stress level decrease, and the curvature of the mean S-N curve decreases as the stress level decreases. The above characteristics can serve as physics knowledge. Many attempts were made to establish the probabilistic fatigue stress-life relationship. They can

be categorized into explicit regression models (e.g., the random fatigue limit model [44]) and machine learning models (e.g., the Probabilistic Physics-guided NN [83]). For the first time, the *P*-*S*-*N* curves are estimated using the Physics-guided Mixture Density Model proposed in this work.

The kernel distribution $\phi_j(\mathrm{y} \mid \mathrm{x})$ in Eq. (3.34) for this problem is expressed as

$$\begin{aligned}\phi_j\left(\log(N_i) \mid \mathbf{x}_i\right) = \delta_i \log\left[f\left(\log(N_i) \mid \mu_j(\mathbf{x}_i), \sigma_j(\mathbf{x}_i)\right)\right] + \\ \left(1-\delta_i\right)\log\left[1 - F\left(\log(N_i) \mid \mu_j(\mathbf{x}_i), \sigma_j(\mathbf{x}_i)\right)\right]\end{aligned}, \tag{3.47}$$

where N is the fatigue life, and the subscripts i and j are the indices for the data point and the component in the mixture distribution, respectively. δ is the index of a failure or runout, and

$$\delta = \begin{cases} 1 & \text{for failure data} \\ 0 & \text{for runout data} \end{cases}.$$

$f(\bullet)$ and $F(\bullet)$ are the probability density function and cumulative distribution function of a Gaussian distribution corresponding to the failure data and runout data, respectively. 1 - $F(\bullet)$ is known as the survival function for the right-censored data [106]. Two physics constraints are imposed on the MDN. They are, as the stress level decreases, (1) the variance of the fatigue life increases, and (2) the curvature of the mean *S*-*N* curve decreases. Constraint (1) is for the first derivative of the mixture variance with respect to the input and is non-positive, and constraint (2) is for the second derivative of the mixture mean with

respect to the input and is non-negative. Those are the inequality constraints in Eq. (3.41). 1 hidden layer with 5 neurons is used. The activation functions are the same as those shown in Fig. 3.15. The penalty parameters r in Eq. (3.43) are 1 initially and increased by a factor of 5 for each of the following iterations, i.e., $r_{k+1} = 5\ r_k$, for both inequality constraints. The initial learning rate lr_0 is 0.01 and $lr_{k+1} = lr_k / 10$ for the following iterations. There are 3 components in the mixture distribution.

The experimental data used for model validation are collected from [44, 108]. The material is Inconel 718. The results are shown in Fig. 3.17 (a). There are 242 failure data ("+" symbols) and 4 runout data (" ▷" symbols). The solid and dashed lines are the fitted mean, and 5% and 95% quantile curves, respectively. Good results can be obtained using the PgMDN model. Both physics constraints (i.e., slope and curvature constraints) are satisfied. It should be noted that a single Gaussian distribution is not appropriate for the data at low strain levels where the mean S-N curve tends to be flat. Using the mixture distribution, those data can be fitted properly. The results shown in Fig. 3.17 (b) are obtained using a single Gaussian distribution. Other than that, the model parameters are the same as those used for Fig. 3.17 (a). In Fig. 3.17 (b) the variance is abnormally large at low strain levels. This is caused by the improperly fitted low bound of the *P-S-N* curves (i.e., 5% quantile curve). Fitting experimental data at low strain levels requires the flat shape of the mean *S-N* curve and large variance. Due to the symmetry of the Gaussian

distribution, using a single Gaussian distribution is difficult for the NN to fit the data well. That problem is solved by using the mixture distribution in this work.

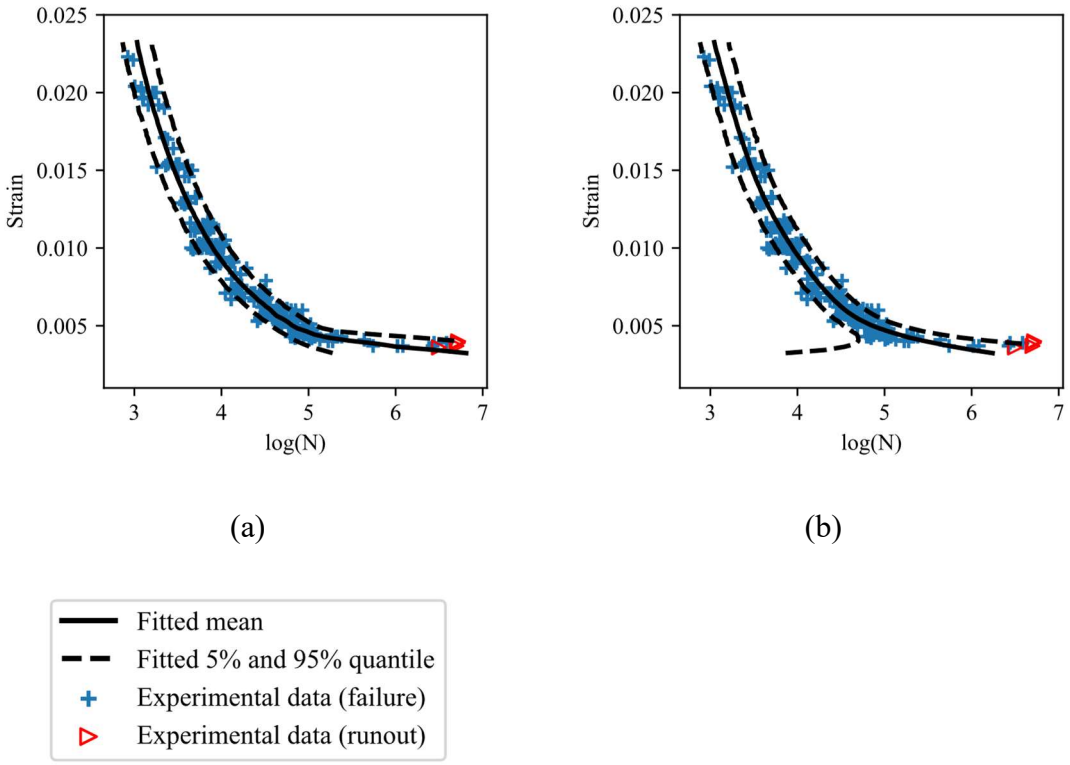


Fig. 3.17 Fitted P-S-N Curves. (a) Number of Mixture Components $k = 3$. (b) $k = 1$.

3.5 Conclusions

Section 3.2 proposed Probabilistic Physics-guided Neural Network (PPgNN) provides a flexible and robust tool for probabilistic fatigue *S-N* estimation. Compared with an explicit regression model, the PPgNN is not restricted to a single function format when fitting with multiple factors and at different stress/strain regimes. It also avoids the non-

physical predictions occurring for the machine learning model and produces both accurate and physically consistent results.

Several major conclusions can be obtained from the current study using investigated data:

- Both the mean and variance can be accurately predicted using the proposed PPgNN architecture with a wide range of fatigue data from both metallic and composite materials. Thus, it appears that the proposed framework is a universal approach not limited to a single material system.
- The proposed new loss function, which integrates both failure and runout data, performs well and shows good convergence behavior for all investigated materials.
- It is shown that the embedded physics guidance is important for fatigue data analysis, especially for extrapolation behavior. The proposed study offers a unique and flexible way to combine data analytics and prior physics knowledge for fatigue data analysis.
- It appears that the relaxation in NN is necessary to reduce the enforced constraints for some materials; it is shown that the relaxation is important for fitting near the fatigue limit region.

- The proposed PPgNN is scalable and can easily extend to include other factors affecting fatigue performance, such as mean stress, frequency, corrosion, etc., by adding additional nodes and layers.

Section 3.3 investigates the modeling for probabilistic fatigue analysis of selective laser melted Ti-6Al-4V considering the effects of process parameters. The Probabilistic Physics-guided Neural Network (PPgNN) is proposed as a modeling tool. With this developed model, both the mean and variance of the fatigue life can be learned. The PPgNN contains proper constraints on model parameters for obtaining P-S-N curves with the physics consistent curvature and nonconstant variance. The PPgNN model is also able to be trained using the data set with missing data for more reliable predictions. Experimental fatigue data are collected from extensive literature for SLM manufactured Ti-64 subjected to various in-process and post-process heat treatment parameters. The PPgNN model is validated using the experimental data.

Section 3.4 presents a model named Physics-guided Mixture Density Network (PgMDN). It integrates a Mixture Density Network for probabilistic modeling and physics knowledge as regularization. This model has no limitations on the distribution types of data. Physics knowledge is used as a constraint. The physics-guided machine learning is modeled as a constrained optimization

problem. Both equality and inequality constraints are considered. The optimization problem is solved by a penalty function method. A dynamic penalty function algorithm is proposed for training the NN. With the physics constraints, the needed data size can be reduced. The validity and application of the PgMDN have been demonstrated using a numerical example and an engineering problem for probabilistic fatigue data analysis. Some discussions are given to illustrate the effectiveness of incorporating the physics knowledge when data are sparse, the improvement of the dynamic penalty function method compared with the static method, and the benefits achieved from the distribution mixture compared with a single Gaussian distribution.

4 PROGNOSTICS BASED ON MULTI-FIDELITY LEARNING

4.1 Overview

Multi-fidelity data exist in almost every engineering and science discipline, which can be from simulation, experiments, and a hybrid form. High-fidelity data are usually associated with higher accuracies and expenses (e.g., high-resolution experimental testing or finer scale simulation), while low-fidelity data are exactly the opposite in terms of accuracy and cost. Multi-fidelity data aggregation (MDA) in this study refers to the process of combining two or multiple sources of different fidelity data to have a high accuracy summary and low computational cost. MDA has a wide range of applications in engineering and science, such as multiscale simulation, multi-resolution imaging, and hybrid simulation-testing. This work presents a novel framework named Multi-fidelity Data Aggregation using Convolutional Neural Networks (MDA-CNN) for multi-fidelity modeling. The MDA-CNN architecture has three components: multi-fidelity data compiling, multi-fidelity perceptive field and convolution, and a deep neural network for mapping. This framework captures and utilizes implicit relationships between any high-fidelity datum and all available low-fidelity data using a defined local perceptive field and convolution. Most existing strategies rely on the collocation method and interpolation, which focuses on the single point relationship. The proposed method has several unique benefits. First, the proposed framework treats the multi-fidelity data as image data and

processes them using CNN, which is very scalable to high dimensional data with more than two fidelities. Second, the flexibility of nonlinear mapping in neural networks facilitates multi-fidelity aggregation and does not need to assume specific relationships among multiple fidelities. Third, the proposed framework does not assume multi-fidelity data are in the same order or from the same physical mechanisms (e.g., the assumption is needed for some error estimation-based multi-fidelity model). Thus, the proposed method can handle data aggregation from multiple sources across different scales, such as different order derivatives and other correlated phenomenon data in a single framework. The proposed MDA-CNN is validated using extensive numerical examples and experimental data with multi-source and multi-fidelity data. Discussions are given to illustrate the benefits and limitations of the proposed framework. Conclusions and future work are presented based on the observations in the current study.

4.2 Multi-fidelity Data Aggregation Using Convolutional Neural Networks

4.2.1 Introduction

In many domains in science and engineering, multiple computational and experimental models are generally available to describe a system of interest. These models differ from each other in the level of fidelity and cost. Typically, computationally or experimentally expensive high-fidelity (HF) models describe the system with high accuracy (e.g., finer scale simulation or high-resolution testing). In contrast, low-fidelity

(LF) models take less time to run but are less accurate [163-165]. Examples of the different levels of fidelities can be simplified/complex mathematical models, coarser/finer discretization of the governing equations, and experimental data with different techniques [166]. In recent years, there have been growing interests in utilizing multi-fidelity (MF) models which combine the advantages of HF and LF models to achieve a required accuracy at a reasonable cost [166]. The approaches to combine fidelities can be categorized into three: adaptation, fusion, and filtering [163]. Adaptation strategy uses adaptation to enhance LF models with information from HF models while the computation proceeds. Fusion approaches evaluate LF models and HF models and then combine information from all outputs. Filtering approaches use the HF model only if the LF model is inaccurate, or when the candidate point meets some criterion.

The concept of multi-fidelity has been explored extensively in surrogate modelings, such as the Gaussian process (GP) [167]. However, limitations of GP in MF modeling still exist, e.g., difficulties during optimization, approximations of discontinuous functions, and high-dimensional problems. On the contrary, neural networks (NN) can deal with arbitrary nonlinearities in high dimensions [83, 84, 168-170]. Recently, efforts of applying neural networks as surrogate models have been made to achieve multi-fidelity. Aydin et al. [171] use cheap low-fidelity computational models to start training the NN and switch to higher-fidelity training data when the overall performance of the NN stops increasing.

Computational models with varying levels of accuracy are needed to generate training data for the NN. This belongs to the filtering strategy. Other works of applying NNs in multi-fidelity problems mainly use adaption. Liu et al. [167] use an LF physics-constrained neural network as the baseline model and use a limited amount of HF data to train a second neural network to predict the difference between the low- and high-fidelity models. Meng et al. [168] propose a composite NN comprised of three NNs. The first NN is trained using the low-fidelity data and coupled to two high-fidelity NNs to discover and exploit nonlinear and linear relationships between low- and high-fidelity data, respectively. Motamed et al. [172] construct a two-level neural network, where a large set of low-fidelity data are utilized to accelerate the construction of a high-fidelity surrogate model with a small set of high-fidelity data.

An important feature of applying NNs to achieve MF modeling is to learn the relationship between low- and high-fidelity models. Current attempts focus on the relationship between HF data and LF data having identical inputs [167, 168, 172]. Thus, a large portion of LF data is not efficiently utilized in the process of learning the appropriate relationships. A novel NN model is proposed named Multi-fidelity Data Aggregation using Convolutional Neural Network (MDA-CNN). It has the following novelties. First, all low-fidelity data are utilized to fully exploit the relationship between low- and high-fidelity models. Second, due to the convolutional operation, this model is very scalable to high-

dimensional data. Third, it is not limited to two levels of fidelity and can be extended to cases with data sets at multiple levels of fidelities. Also, the proposed framework does not assume multi-fidelity data are in the same order or from the same physical mechanisms (e.g., the assumption is needed for some error estimation-based multi-fidelity model). Thus, the proposed method can handle data aggregation from multiple sources across different scales, such as different order derivatives and other correlated phenomenon data in a single framework. Fourth, it is an integrated NN rather than a composite of several NNs. Thus, only one-time training is needed, and training time is shortened.

4.2.2 Multi-fidelity Data Aggregation Using Convolutional Neural Networks

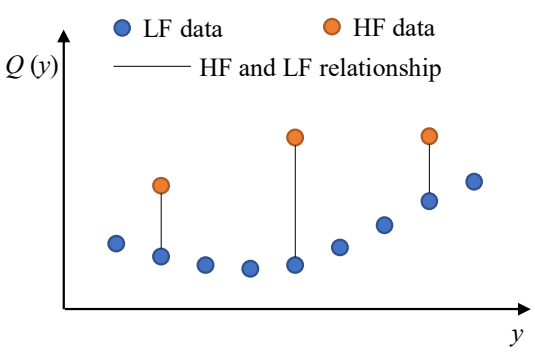
Suppose an n-dimensional random vector $y \in \mathbb{R}^n$ is mapped through a model to obtain a desired output quantity $Q(y) \in \mathbb{R}$. Let $Q_L(y)$ and $Q_H(y)$ denote the approximated values of the quantity $Q(y)$ by a low- and high-fidelity computational model, respectively. Consider a general relationship between the two models as

$$Q_H(y) = F\left(y, Q_L(y)\right), \tag{4.1}$$

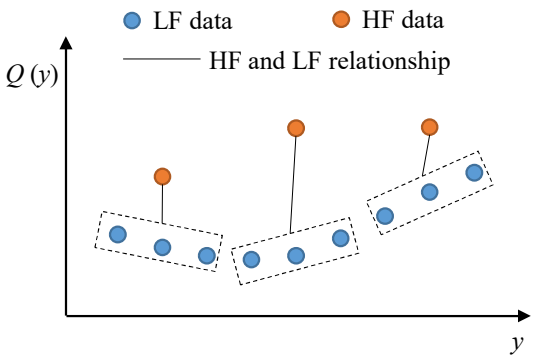
where $F(\bullet)$ is an unknown function that captures the relationship between low- and high-fidelity quantities. $F(\bullet)$ can be either linear or nonlinear [168, 172].

The relationship in Eq. (4.1) can be presented in Fig. 4.1 (a) when y is one dimensional. For each high-fidelity data point $Q_H(y_H)$ at y_H, there exists a corresponding low-fidelity $Q_L(y_H)$ data point. The multi-fidelity strategy is to capture the relationship

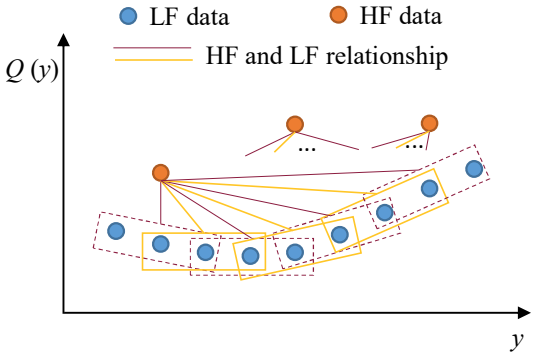
between Q_H (y_H) and Q_L (y_H). To learn the relationship F (•), a neural network surrogate model is constructed. The architecture of the NN model is shown in Fig. 4.2. There are two input neurons (shown in blue color) in the input layer. They are y_H and Q_L (y_H). The output neuron in the output layer (shown in yellow color) is Q_H (y_H). The hidden layers between the input and output layer are composed of two parts: linear mapping and nonlinear mapping. The linear and nonlinear mapping are learned through a skip connection (the upper arrow) and fully connected layers (the lower arrow), respectively. The linear and nonlinear mapping are then added together before being connected to the output layer. The neurons in the hidden layers with grey and green color use linear and nonlinear activation functions, respectively. The decomposition of the hidden layers into linear and nonlinear parts is inspired by the concept of ResNet [173] where a direct link is used to learn the residual rather than the original function. It is shown that it is easier to optimize the residual mapping than to optimize the original one. If there are no more things the nonlinear fully connected layers can learn, it just learns the mapping as being zero. It turns out that it is easier for the network to learn a mapping closer to zero than a linear mapping [173, 174].



(a) Relationship between HF data and corresponding LF data.



(b) Relationship between HF data and neighboring LF data.



(c) Relationship between HF data and every LF datum.

Fig. 4.1 Relationships Between HF Data and LF Data.

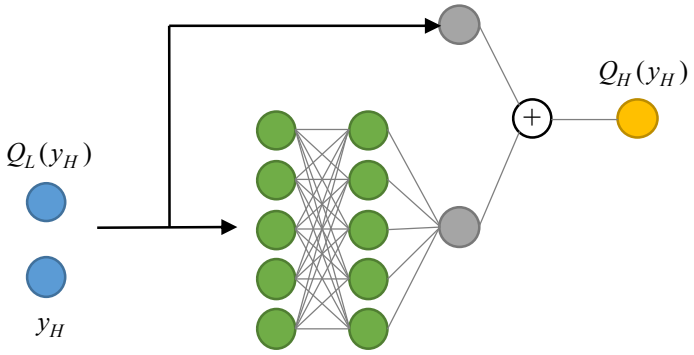


Fig. 4.2 Neural Network Architecture for Capturing the Relationship Between HF Data and Corresponding LF Data (Fig. 1(a))

The aforementioned multi-fidelity model learns the relationship between low- and high-fidelity data at the same inputs, i.e., a point-to-point relationship (see Fig. 4.1 (a)). With that multi-fidelity strategy, the utilized low-fidelity data are limited to the data with the same inputs as the corresponding high-fidelity data at y_H. A large portion of low-fidelity data is not effectively used to learn the relationship. In order to take those unused LF data into account, A natural (also the proposed) idea is schematically presented in Fig. 4.1 (b). The proposed idea is to learn the relationship between high-fidelity data and low-fidelity data in the neighborhood of y_H (shown in the dashed line box in Fig. 4.1 (b)). In this way, the multi-fidelity model can learn the point-to-domain relationship. Thus, more information from a low fidelity model can be incorporated to assist the multi-fidelity strategy. More generally, we can expand the above idea to learn the relationship between a high-fidelity datum and all available low-fidelity data as shown in Fig. 4.1 (c). That is

achieved by moving the local domain (dashed line box in Fig. 4.1 (c)) in the space of low-fidelity data sequentially. This sliding window operation captures the relationship between each high-fidelity data and each local domain of low-fidelity data. It is interesting to see that this "sliding window" operation is an analogy to the local receptive field concept and convolutional operation. This is the reason why we use CNN to do multi-fidelity data analytics. The remains of this section describe the implementation of the idea in Fig. 4.1 (c) through a proposed neural network model: Multi-fidelity Data Aggregation using Convolutional Neural Networks (MDA-CNN).

The architecture of MDA-CNN is shown in Fig. 4.3. It is composed of three parts: multi-fidelity data compiling, convolutional layer, and deep neural network. Compared with the architecture in Fig. 4.2, the output data is still $Q_H(y_{H,i})$, $i = 1,\ldots, N_H$, where N_H is the number of high-fidelity data. Instead of just inputting $y_{H,i}$ and $Q_L(y_{H,i})$, $i = 1,\ldots, N_H$, all the low-fidelity data $y_{L,j}$ and $Q_L(y_{L,j})$, $j = 1,\ldots, N_L$ are also used as inputs, where N_L is the number of low-fidelity data. The input data are compiled in the form of a table shown in the first dashed line box in Fig. 4.3. In this way, we have a total number of N_H input tables. In any ith input table, $i = 1,\ldots, N_H$, there are four columns: y_L, $Q_L(y_L)$, $y_{H,i}$ and $Q_L(y_{H,i})$. There are a total number of N_L rows. For the first two columns, y_L and $Q_L(y_L)$, the table values are $y_{L,j}$, $Q_L(y_{L,j})$, $j = 1,\ldots, N_L$, respectively, for the jth row. For the last two columns, the table values are $y_{H,i}$ and $Q_L(y_{H,i})$, respectively. They are the same for each row. Being

complied in the above manner, all the available low-fidelity data are utilized for the NN model to learn the relationship between low- and high-fidelity data.

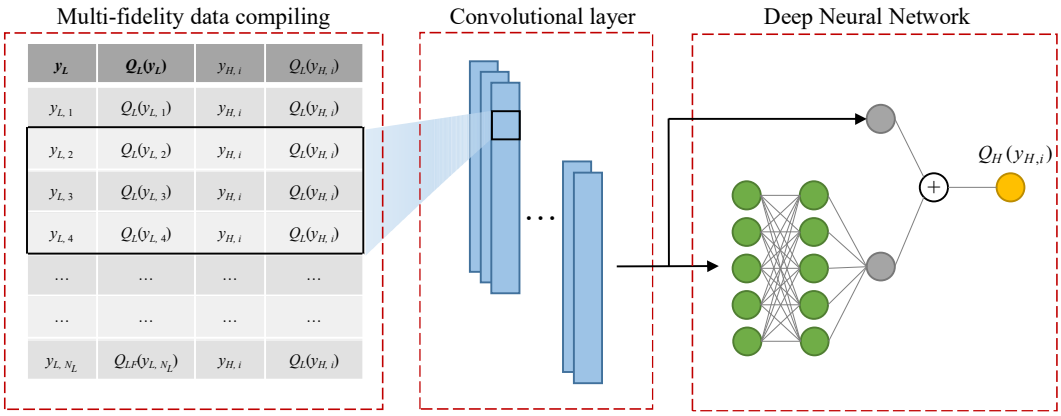


Fig. 4.3 Multi-fidelity Data Aggregation Using Convolutional Neural Networks.

The input is a matrix in Fig. 4.3 instead of a vector in Fig. 4.2. Analogous to an image, the input table is then connected to the convolutional layer. The black rectangular window shown in the input table is a local receptive field. As an illustration in Fig. 4.3, it is a 3 × 4 region, which can be understood as learning the relationship between $Q_H(y_{H,i})$ and the integration of $Q_L(y_{H,i})$ at $y_{H,i}$, $Q_L(y_{L,2})$ at $y_{L,2}$, $Q_L(y_{L,3})$ at $y_{L,3}$, and $Q_L(y_{L,4})$ at $y_{L,4}$. This operation corresponds to the connection of a high-fidelity datum with a local domain of low-fidelity data shown in Fig. 4.1 (c). Each row of the input table is treated as a unit. The local receptive field can be composed of one unit or multiple units. Each local receptive field is connected to a hidden neuron in the convolutional layer. Next, a new local receptive field is generated by sliding down the previous one by a row. That is the fulfillment of

moving the local domain of low-fidelity data (dashed line box) to the right by one low-fidelity data shown in Fig. 4.1 (c). Each local receptive field is connected to a different hidden neuron in the convolutional layer. This procedure is conducted across the entire input table. A feature map (a blue rectangle in the second dashed line box in Fig. 4.3) from the input layer to the convolutional layer is constructed. The feature map can detect a single type of localized feature of the relationship between low- and high-fidelity data. To learn a complete relationship, more than one feature map is needed. Thus, a complete convolutional layer consisting of multiple different feature maps [175] is constructed and is shown in the second dashed line box in Fig. 4.3. Next, a deep neural network shown in the third dashed line box in Fig. 4.3 is used to learn the nonlinear mapping, which is the same as the architecture in Fig. 4.2 without the input layer.

The MDA-CNN in Fig. 4.3 is for the bi-fidelity problem with one-dimensional y. It is noted that the proposed framework is not limited to two levels of fidelity and can be extended to process data sets at multiple levels of fidelities and to deal with y with any dimension. In addition, if other information besides Q_L (y) from a low-fidelity model is useful, such as derivatives, it is also easy to be incorporated into this framework. For those scenarios, what one needs to do is just to adapt the multi-fidelity data compiling in the first dashed line box in Fig. 4.3 while leaving the MDA-CNN framework unchanged. Fig. 4.4 shows the design of the input table for scenarios of two-dimension y, two low-fidelity

models, and utilizing first derivatives of the low-fidelity model, respectively. The basic idea for constructing the input table is that the left half of the table contains all the low-fidelity information, and the right half indicates the low-fidelity datum corresponding to an available high-fidelity datum. Some of these multi-source and multi-fidelity data compiling and learning are illustrated later in this section.

y_{L1}	y_{L2}	$Q_L(y_{L1}, y_{L2})$	$y_{H1,i}$	$y_{H2,i}$	$Q_L(y_{H1,i}, y_{H2,i})$
$y_{L1,1}$	$y_{L2,1}$	$Q_L(y_{L1,1}, y_{L2,1})$	$y_{H1,i}$	$y_{H2,i}$	$Q_L(y_{H1,i}, y_{H2,i})$
$y_{L1,2}$	$y_{L2,2}$	$Q_L(y_{L1,2}, y_{L2,2})$	$y_{H1,i}$	$y_{H2,i}$	$Q_L(y_{H1,i}, y_{H2,i})$
$y_{L1,3}$	$y_{L2,3}$	$Q_L(y_{L1,3}, y_{L2,3})$	$y_{H1,i}$	$y_{H2,i}$	$Q_L(y_{H1,i}, y_{H2,i})$
$y_{L1,4}$	$y_{L2,4}$	$Q_L(y_{L1,4}, y_{L2,4})$	$y_{H1,i}$	$y_{H2,i}$	$Q_L(y_{H1,i}, y_{H2,i})$
…	…	…	…	…	…
…	…	…	…	…	…
y_{L1,N_L}	y_{L2,N_L}	$Q_L(y_{L1,N_L}, y_{L2,N_L})$	$y_{H1,i}$	$y_{H2,i}$	$Q_L(y_{H1,i}, y_{H2,i})$

(a) Two-dimension *y*.

y_L	$Q_{L1}(y_L)$	$Q_{L2}(y_L)$	$y_{H,i}$	$Q_{L1}(y_{H,i})$	$Q_{L2}(y_{H,i})$
$y_{L,1}$	$Q_{L1}(y_{L,1})$	$Q_{L2}(y_{L,1})$	$y_{H,i}$	$Q_{L1}(y_{H,i})$	$Q_{L2}(y_{H,i})$
$y_{L,2}$	$Q_{L1}(y_{L,2})$	$Q_{L2}(y_{L,2})$	$y_{H,i}$	$Q_{L1}(y_{H,i})$	$Q_{L2}(y_{H,i})$
$y_{L,3}$	$Q_{L1}(y_{L,3})$	$Q_{L2}(y_{L,3})$	$y_{H,i}$	$Q_{L1}(y_{H,i})$	$Q_{L2}(y_{H,i})$
$y_{L,4}$	$Q_{L1}(y_{L,4})$	$Q_{L2}(y_{L,4})$	$y_{H,i}$	$Q_{L1}(y_{H,i})$	$Q_{L2}(y_{H,i})$
…	…	…	…	…	…
…	…	…	…	…	…
y_{L,N_L}	$Q_{L1}(y_{L,N_L})$	$Q_{L2}(y_{L,N_L})$	$y_{H,i}$	$Q_{L1}(y_{H,i})$	$Q_{L2}(y_{H,i})$

(b) Two low-fidelity models.

y_L	$Q_L(y_L)$	$Q_L^{(1)}(y_L)$	$y_{H,i}$	$Q_L(y_{H,i})$	$Q_L^{(1)}(y_{H,i})$
$y_{L,1}$	$Q_L(y_{L,1})$	$Q_L^{(1)}(y_{L,1})$	$y_{H,i}$	$Q_L(y_{H,i})$	$Q_L^{(1)}(y_{H,i})$
$y_{L,2}$	$Q_L(y_{L,2})$	$Q_L^{(1)}(y_{L,2})$	$y_{H,i}$	$Q_L(y_{H,i})$	$Q_L^{(1)}(y_{H,i})$
$y_{L,3}$	$Q_L(y_{L,3})$	$Q_L^{(1)}(y_{L,3})$	$y_{H,i}$	$Q_L(y_{H,i})$	$Q_L^{(1)}(y_{H,i})$
$y_{L,4}$	$Q_L(y_{L,4})$	$Q_L^{(1)}(y_{L,4})$	$y_{H,i}$	$Q_L(y_{H,i})$	$Q_L^{(1)}(y_{H,i})$
…	…	…	…	…	…
…	…	…	…	…	…
y_{L,N_L}	$Q_L(y_{L,N_L})$	$Q_L^{(1)}(y_{L,N_L})$	$y_{H,i}$	$Q_L(y_{H,i})$	$Q_L^{(1)}(y_{H,i})$

(c) Utilizing first derivatives of the low-fidelity model.

Fig. 4.4 The Design of the Input Table for Multi-source and Multi-fidelity Data Aggregation.

4.2.3 Numerical Experiments and Validation

Seven numerical examples are adopted for validating the MDA-CNN model. They are continuous functions with linear relationship [168], discontinuous functions with linear relationship [168], continuous functions with nonlinear relationship, continuous oscillation functions with nonlinear relationship [168], phase-shifted oscillations [168], different periodicities [176], and 50-dimensional functions [168] as shown in Table 4.1 (a) – (g), respectively. The results from a neural network using only high-fidelity data are presented in Ref. [168] for Example (a), (b), (d), (e), and (g). It was shown that the predictions are inaccurate due to the lack of high-fidelity data.

The architecture in Fig. 4.3 is used to obtain the multi-fidelity results. For Example (a) - (d) the input table is organized as the one shown in Fig. 4.3. For Example (e) and (f), the input table shown in Fig. 4.4 (c) is employed. The first derivative is calculated using the central difference method with step 10^{-6}. Example (g) uses the input table shown in Fig. 4.4 (a). The hyperparameters used for training the MDA-CNN model are listed in Table 4.2. For each numerical example, the Adam optimization is employed to minimize the mean squared error. 2 hidden layers with 10 neurons per layer and hyperbolic tangent

activation functions are employed for the fully connected layers. Table 4.2 also shows the number of low- and high-fidelity data used for training the MDA-CNN model for each example. For Example (a) – (f), the low- and high-data points are uniformly selected, and for Example (g), they are randomly selected.

The results of the seven numerical examples are shown in Fig. 4.5 (a) – (g), respectively. They are the comparison between the results from the MDA-CNN model and ground truth high fidelity models. The smaller blue points and bigger red points are the low- and high-fidelity data used for training, respectively. The blue, orange and green lines are the plots from the low-, high- and multi-fidelity model. It can be seen that the results from the MDA-CNN are almost identical to those from pure high-fidelity models for Example (a) – (f). For Example (g), the predictions are made at 10,000 random locations. The predicted vs. actual value is plotted in Fig. 4.5 (g). The points are almost located on the solid line representing exact predictions. The prediction errors calculated by

$$Error = (Q_{predicted} - Q_{actual}) / Q_{actual} \tag{4.2}$$

are 0.0035, 0.0027, and 0.0179 for mean, standard deviation, and maximum value, respectively. Therefore, good accuracy can be achieved for all the numerical examples investigated.

Table 4.1 Low- and High-fidelity Models for Numerical Examples.

(a)	Continuous functions with the linear relationship

LF model
$$Q_L(y) = 0.5(6y-2)^2 \sin(12y-4), \quad +10(y-0.5)-5 \qquad 0 \le y \le 1 \tag{4.3}$$

HF model
$$Q_H(y) = (6y-2)^2 \sin(12y-4), \qquad 0 \le y \le 1 \tag{4.4}$$

Discontinuous functions with the linear relationship

(b)

LF model
$$Q_L(y) = \begin{cases} 0.5(6y-2)^2 \sin(12y-4) + 10(y-0.5), & 0 \le y \le 0.5 \\ 3 + 0.5(6y-2)^2 \sin(12y-4) + 10(y-0.5), & 0.5 < y \le 1 \end{cases} \tag{4.5}$$

HF model
$$Q_H(y) = \begin{cases} 2Q_L(y) - 20y + 20, & 0 \le y \le 0.5 \\ 4 + 2Q_L(y) - 20y + 20, & 0.5 < y \le 1 \end{cases} \tag{4.6}$$

Continuous functions with the nonlinear relationship

(c)

LF model
$$Q_L(y) = 0.5(6y-2)^2 \sin(12y-4) + 10(y-0.5) - 5 \tag{4.7}$$

HF model
$$Q_H(y) = (6y-2)^2 \sin(12y-4) - 10(y-1)^2 \tag{4.8}$$

Continuous oscillation functions with the nonlinear relationship

(d)

LF model
$$Q_L(y) = \sin(8\pi y), \quad 0 \le y \le 1 \tag{4.9}$$

HF model
$$Q_H(y) = (y - \sqrt{2}) Q_L^2(y), \quad 0 \le y \le 1 \tag{4.10}$$

Phase-shifted oscillations

(e)

LF model
$$Q_L(y) = \sin(8\pi y) \tag{4.11}$$

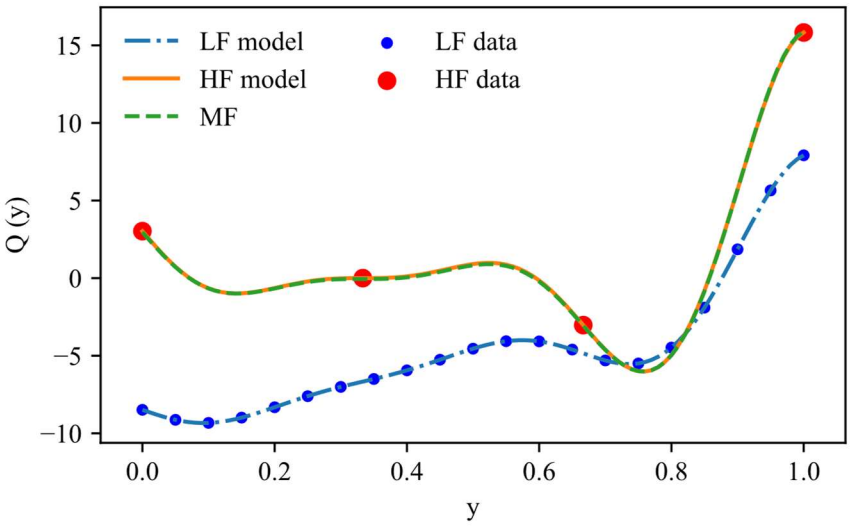
HF model
$$Q_H(y) = y^2 + Q_L^2(y + \pi/10) \tag{4.12}$$

(f) Different periodicities

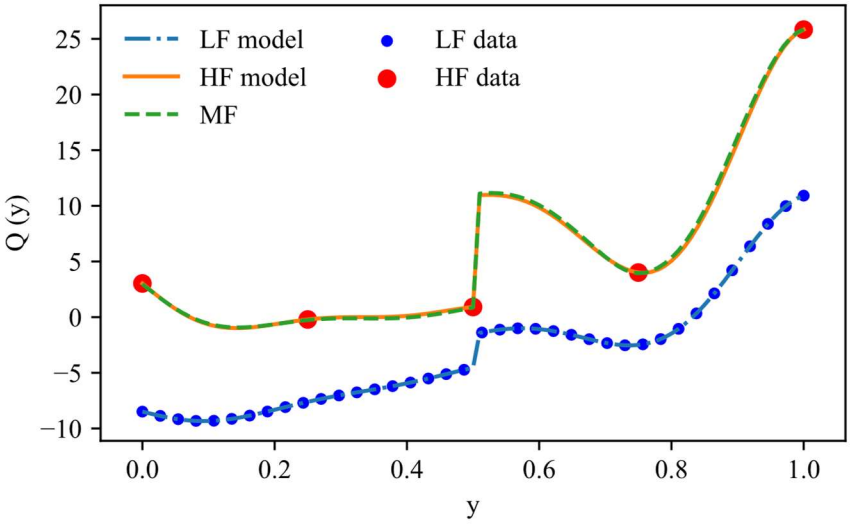
	LF model	$Q_L(y) = \sin(6\sqrt{2}\pi y)$	(4.13)
	HF model	$Q_H(y) = \sin(8\pi y + \pi/10)$	(4.14)
		50-dimensional functions	
(g)	LF model	$Q_L(y) = 0.8Q_H(y) - \sum_{i=1}^{49} 0.4 y_i y_{i+1} - 50, \quad -3 \le y_i \le 3$	(4.15)
	HF model	$Q_H(y) = (y_1 - 1)^2 + \sum_{i=2}^{50} (2y_i^2 - y_{i-1})^2, \quad -3 \le y_i \le 3$	(4.16)

Table 4.2 Hyperparameters and Number of Low- and High-fidelity Data Used While Training the MDA-CNN for Numerical Examples.

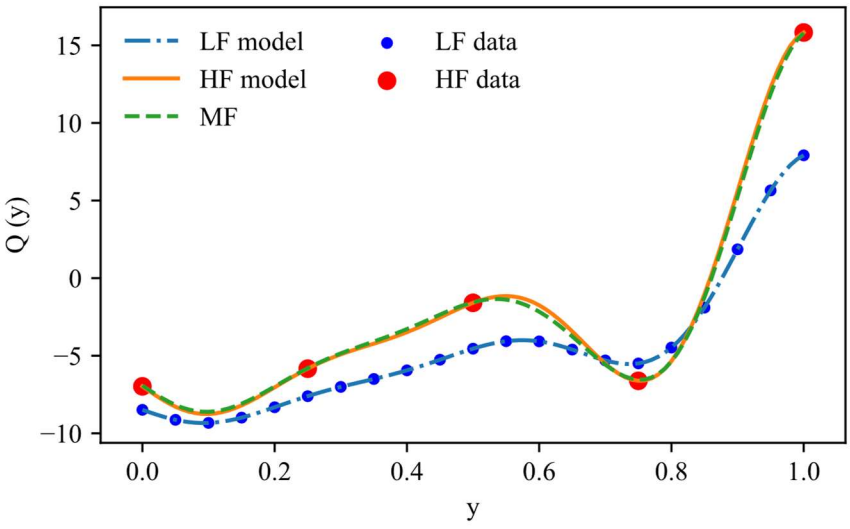
	(a)	(b)	(c)	(d)	(e)	(f)	(g)
Number of epochs	5,000	5,000	5,000	5,000	5,000	5,000	1,000
Batch size	4	5	5	10	10	10	50
Learning rate	0.001						
Regularization rate	0.01						
Number of feature maps	64						
Kernel width	3						
Number of LF data	21	38	21	51	51	51	1,000
Number of HF data	4	5	5	15	16	15	100



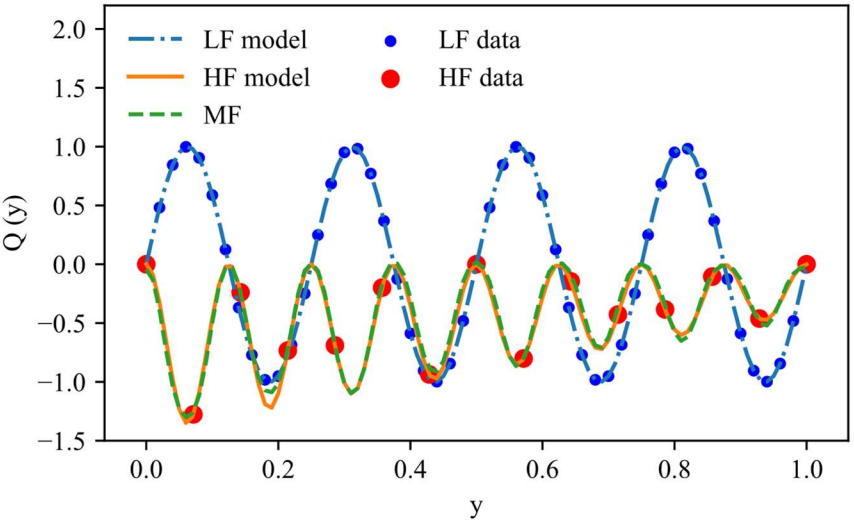
(a) Continuous function with the linear relationship.



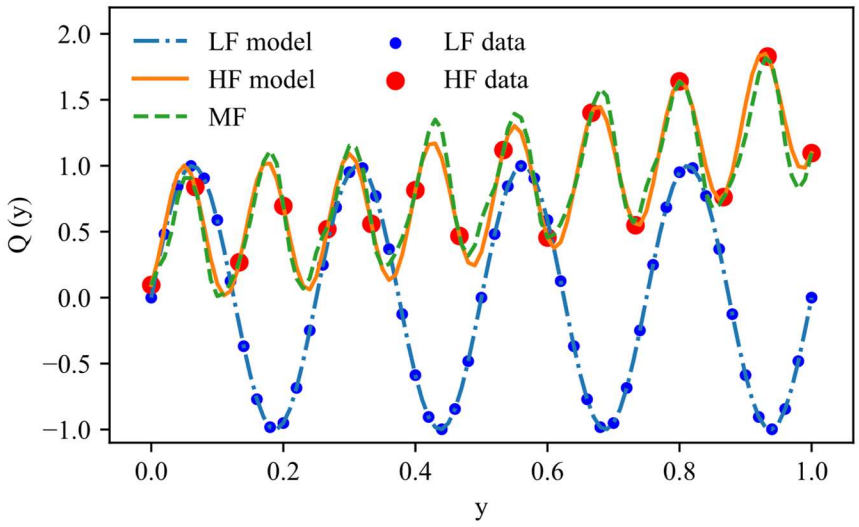
(b) Discontinuous function with the linear relationship.



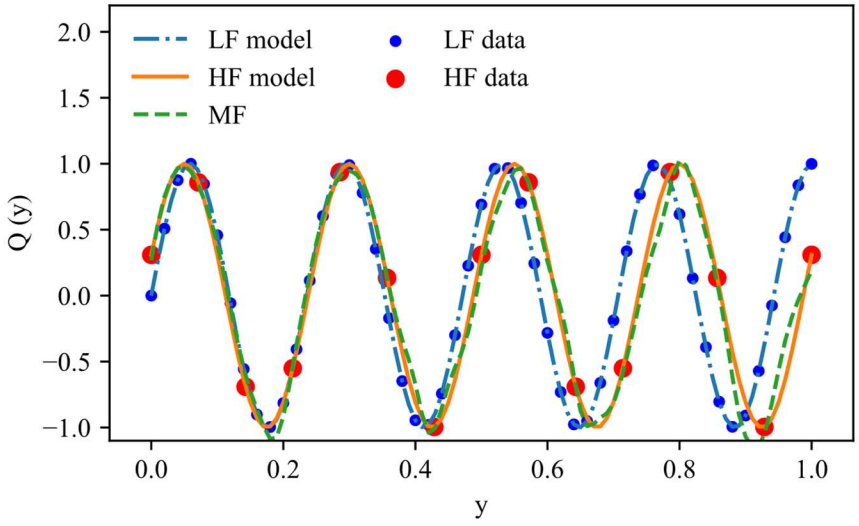
(c) Continuous function with the nonlinear relationship.



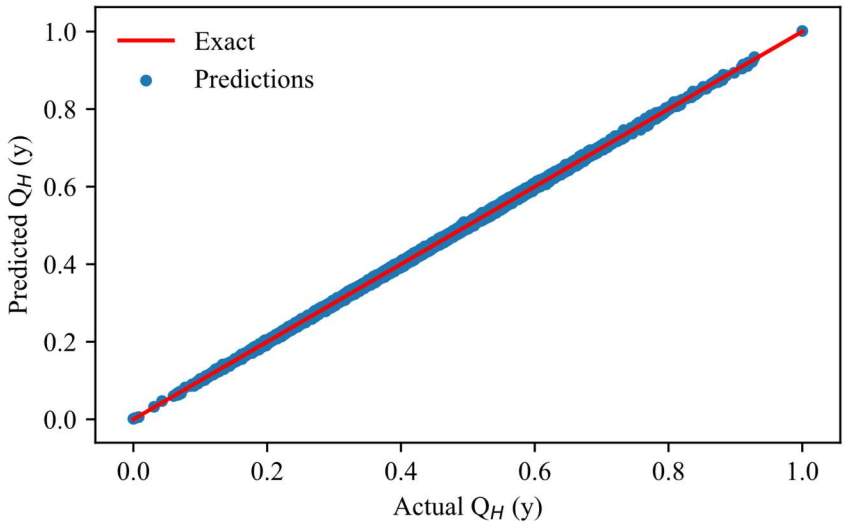
(d) Continuous oscillation function with the nonlinear relationship.



(e) Phase-shifted oscillations.



(f) Different periodicity.



(g) 50-dimensional function.

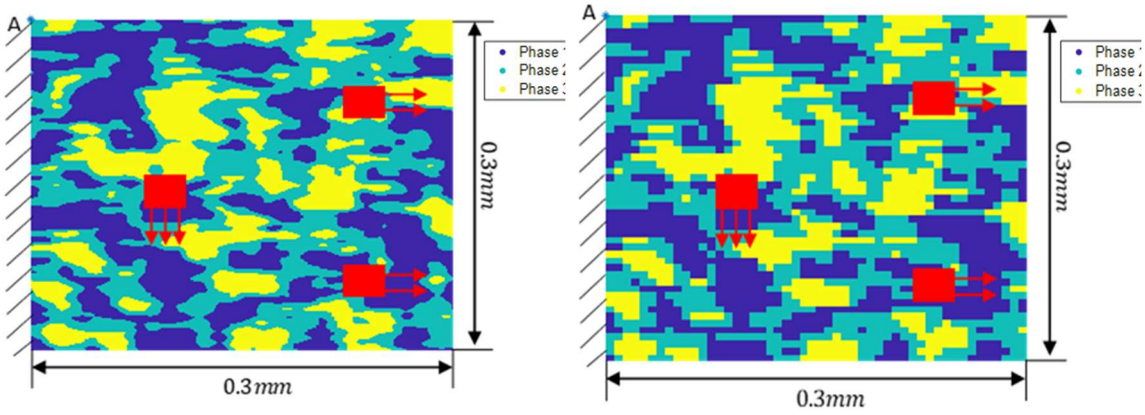
Fig. 4.5 Approximations from Multi-fidelity Data Aggregation Using Convolutional Neural Networks (MDA-CNN) for Numerical Examples in Table 4.1.

4.2.4 Engineering Application Examples: Finite Element Analysis with Random Microstructure

Consider a two-dimensional (2D) 0.3 mm × 0.3 mm plate consisting of three-phase heterogenous materials shown in Fig. 4.6. Different colors indicate different phases of materials. The randomness in microstructures can affect the overall performance of small devices or components such as microelectromechanical systems (MEMS) [177] due to the compatible scale. Thus, stress analysis due to the existence of microstructures is essential for either failure analysis or design. The microstructure is generated by using a recently developed Mixture Random Field model [178] and is kept unchanged in all simulations. The high-fidelity model with a mesh of 300 × 300 is illustrated in Fig. 4.6 (a) which can precisely represent the details of the microstructure. Fig. 4.6 (b) is the low-fidelity model

which only has a mesh of 50 × 50. Some of the microstructure details are lost due to the coarse mesh. The nodes on the left edge are fixed in both the x and y-direction. External body forces in either x or y direction with the amplitude of 1 kN/mm^2 are distributed in the red rectangular areas (see Fig. 4.6). The red arrows in Fig. 4.6 show the directions of the applied forces. The structure is assumed to be under plane stress.

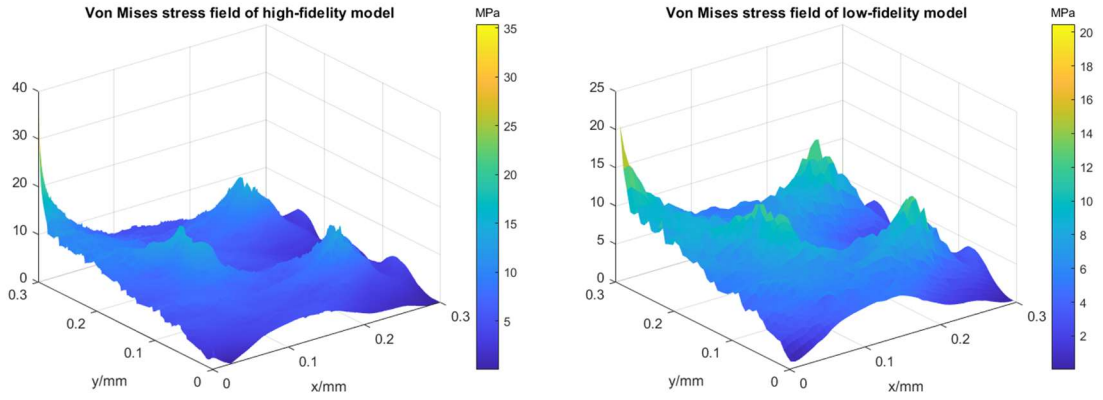
To avoid the high computational costs for probabilistic analysis or design where repeated response function calls are needed, the numerically efficient multi-fidelity model is trained to learn the mapping from different material properties to responses of critical points. Young's moduli of the three materials are chosen as random variables for illustration purposes. Thus, the inputs are three-dimensional. The comparison between the von Mises stress fields calculated by the low- and high-fidelity model is presented in Fig. 4.7 with the input vector of (160, 190, 230) GPa for Young's modulus of the three-phase material. For both fidelity models, the stress concentration occurs at point A at the top left corner with the coordinate of (0, 0.3) mm. Point A can be regarded as the most dangerous point. Thus, the von Mises stress at that point is selected to be the output of the multi-fidelity model. It should be noted that any other location outputs can be selected, and the proposed method is not limited to the location in the simulation domain.



(a) High-fidelity model, 300 × 300.

(b) Low-fidelity model, 50 × 50.

Fig. 4.6 The Microstructure of the 2D Plate, Applied Forces, and Boundary Conditions.



(a) High-fidelity model, 300 × 300.

(b) Low-fidelity model, 50 × 50.

Fig. 4.7 The Von Mises Stress Field of (a) High-fidelity and (b) Low-fidelity Model with (160, 190, 230) GPa for Young's Modulus of the Three-phase Material.

The three-dimensional input space for the low-fidelity (LF) model is uniformly selected from intervals of [150, 170], [180, 200], and [210, 250] GPa for three materials, respectively. The grid length (i.e., the distance between two adjacent points) in each

interval is 5 GPa. The total number of LF data is 255. The comparison between low- and high-fidelity results is shown in Fig. 4.8. The results are normalized by subtracting the minimum of the high-fidelity results and then being divided by the difference between the maximum and minimum of the high-fidelity results. It can be seen that the low-fidelity results do not agree with the high-fidelity results. The root-mean-square error is 1840.3% as shown in Table 4.3. However, the overall trend of the low- and high-fidelity results match. That is, in general, the high-fidelity results increase as the low-fidelity results increase.

Next, the multi-fidelity method is utilized for a more accurate prediction. The input space for the high-fidelity (HF) model is the grid from input vectors (155, 165), (185, 195), and (220, 230, 240) GPa. The total number of HF data is 12. The architecture of Multi-fidelity Data Aggregation using Convolutional Neural Networks (MDA-CNN) in Fig. 4.3 is used for the multi-fidelity modeling. Since the input of this problem is three-dimensional, the input table is designed according to Fig. 4.4 (a). There are 64 feature maps in the convolutional layer. 2 hidden layers with 10 neurons per layer and hyperbolic tangent activation functions are employed for the fully connected layers. Predictions are made at the 255 LF training data. The results from the MDA-CNN are shown in Fig. 4.9. The green circles are the predictions, and the black solid line represents the predictions equal to the ground truth (high-fidelity results). The agreement between the green circles and the black line shows that good predictions can be obtained using the MDA-CNN. The RMSE is 1.2% as shown in Table 4.3. To show the necessity of multi-fidelity modeling, a neural network is trained for prediction only using the 12 HF data. That NN is fully connected with 2

hidden layers each of which has 10 neurons. The results are shown in blue triangles in Fig. 4.9 and the RMSE is 38.9% in Table 4.3. Thus, using single-fidelity modeling is insufficient due to the insufficiency of HF data. To show the effectiveness of incorporating the convolutional layer in the MDA-CNN to learn the relationship between an HF datum and all the available LF data, the predictions are also made using the multi-fidelity neural network architecture in Fig. 4.2 which only learns the relationship between HF data and corresponding LF data. There is no convolutional layer in this architecture. Other than that, the architecture is the same as the MDA-CNN used in this problem for a fair comparison. The results are shown in pink squares in Fig. 4.9 and the RMSE is 18.6% in Table 4.3. Predictive improvement can be achieved compared with single-fidelity modeling. However, without incorporating the convolutional layer, the prediction accuracy is still poorer than those obtained with the MDA-CNN.

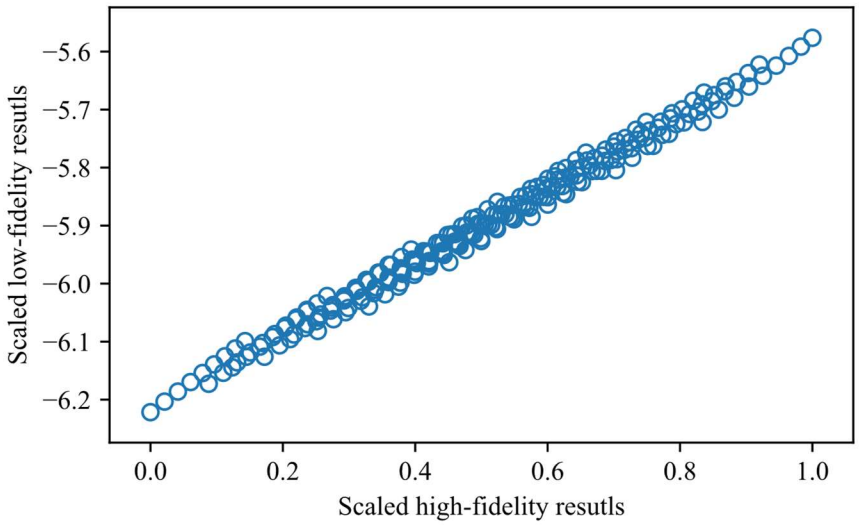


Fig. 4.8 Comparison Between Low- and High-fidelity Results.

Table 4.3 RMSE (%) for Different Models.

	LF	Single-fidelity NN	MFNN	MDA-CNN
RMSE	1840.3	38.9	18.6	1.2

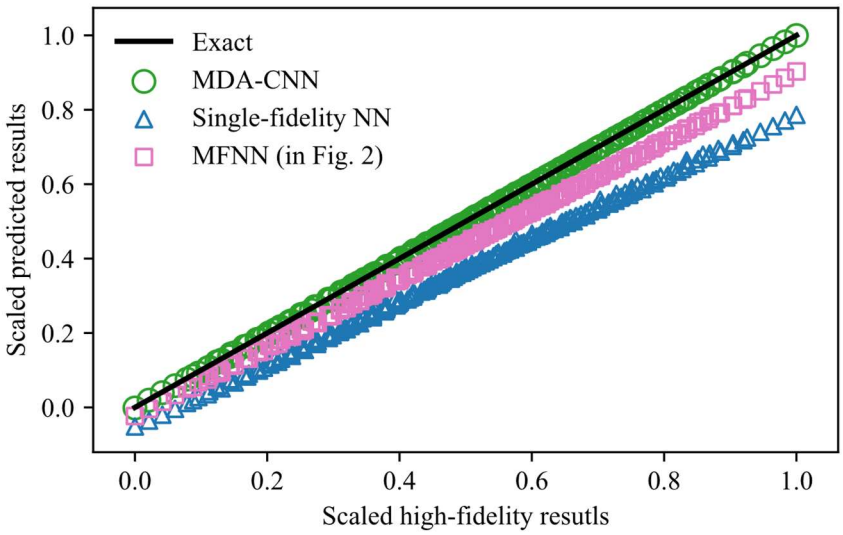


Fig. 4.9 Comparison of Predicted Results with High-fidelity Results.

Next, the computational cost for this problem is discussed. Let $W_{\text{MDA-CNN}}$ and W_{HF} denote the total computational cost for multi-fidelity modeling and the classical high-fidelity model, respectively. They can be expressed in the following forms,

$$W_{\text{MDA-CNN}} = N_L w_L + N_H w_H + w_T + w_P \,, \tag{4.17}$$

and

$$W_{\text{HF}} = N_P w_H \,, \tag{4.18}$$

where w_L and w_H are the computational cost for obtaining Q_L and Q_H from a low- and high-computational model, respectively, N_L and N_H are the numbers of low- and high-fidelity data for training, respectively, w_T and w_P are the computational cost for training and evaluating the MDA-CNN, respectively, and N_P is the number of evaluations (predictions) for the MDA-CNN or HF model. The cost w_T for training the NN depends on the size of training data and NN architecture (i.e., the number of hidden layers and neurons, etc.). It is a one-time cost. The cost w_P for evaluating the NN depends on the NN architecture. It involves activation function and matrix-operation [172]. It is observed for this problem that this cost is almost independent of the number of predictions. For this problem, the computational costs and data sizes are shown in Table 4.4. The computational cost vs. the number of evaluations is plotted in Fig. 4.10 according to Eqs. (4.17) and (4.18). When the number of evaluations exceeds 22, the more evaluations, the more computational saving can be achieved by using the MDA-CNN. It should be noted that this overhead suggests that for the linear low-dimensional problem, MDA-CNN does not offer computational advantages as very few function evaluations of the high-fidelity model is sufficient. However, MDA-CNN has superior efficiency for most engineering applications with nonlinearity and high-dimensionality.

Table 4.4 Computational Costs and Data Sizes.

w_L (second)	w_H (second)	w_T (second)	w_P (second)	N_L	N_H
0.096	2.968	5.075	0.054	255	12

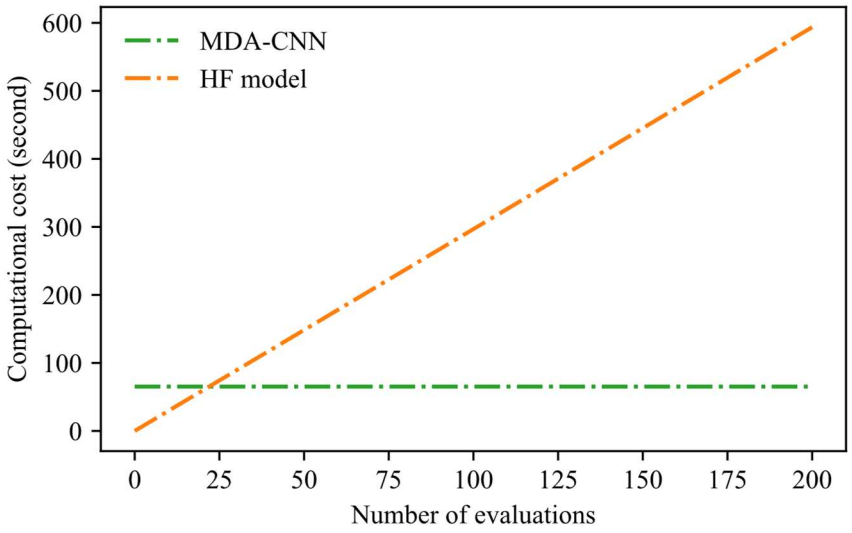


Fig. 4.10 Comparison of Computational Cost Between the MDA-CNN and HF Model.

4.2.5 Fatigue Crack Growth Prognosis with Monitoring

In prognostics for engineering materials and systems, both simulation models and experimental measurements are available. Experimental measurements can be used to update the simulation model for more accurate remaining life prediction [28, 49, 141, 179]. In this application, data from simulation models are relatively easy to obtain as the computational complexity is usually not high. Experimental measurements are usually very expensive but represent the true response from materials and structures. Thus, we treat the simulation data as the low-fidelity data and the experimental measurements as the high-fidelity data. The proposed multi-fidelity data aggregation can be applied to predict the crack growth trajectory under fatigue loadings [28, 180].

An aluminum 2024-T3 plate with an initial center-through crack is shown in Fig. 4.11. The plate has dimensions of width w = 152.4 mm, length L = 558.8 mm, and thickness t = 2.54 mm. The initial crack size is a_0 = 9.0 mm. The cyclic loading is applied with a

stress amplitude of 24.14 MPa, frequency of 20 Hz, and stress ratio of 0.2. 68 plates were tested with the same specimen and loading configurations. The experimental data of crack growth trajectories (crack growth vs. loading cycles) were reported [180]. Those trajectories vary from each other due to the material and loading uncertainty.

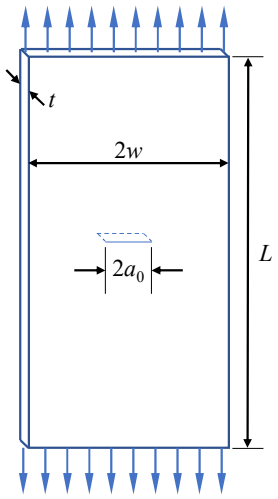


Fig. 4.11 An Aluminum 2024-T3 Plate with an Initial Center-through Crack Under Fatigue Loading.

The multi-fidelity problem setup is as follows. A simplified mathematical model calibrated by historical data is treated as the low-fidelity model. Specifically, 10 out of 68 trajectory data are used to fit parameters of the Paris model (a simplified mathematical model). The fitted model cannot precisely predict the crack growth trajectory of a new specimen due to the probabilistic nature of fatigue [9, 16, 83, 88, 162, 181]. However, that model can describe the approximate trend of crack growth trajectories under repeated testing circumstances. Thus, it is used as the low-fidelity model. From the remaining dataset, one trajectory is arbitrarily selected as the result of new testing, which is treated as

the high-fidelity model to be predicted. Three crack size measurements at earlier stages from that trajectory representing the actual inspection data are used as high-fidelity data. The complete crack growth trajectory is predicted with the low-fidelity model and sparse high-fidelity data.

The Paris' model [182] is expressed as

$$\frac{da}{dN} = c\left(\Delta K\right)^m, \tag{4.19}$$

where a is the crack length, N is the number of applied loading cycles, and c and m are material parameters. ΔK is stress intensity variation and is calculated by

$$\Delta K = \Delta\sigma\sqrt{\pi a}\sqrt{\sec(\pi a / w)}. \tag{4.20}$$

Using the 10 trajectory data (historical data), model parameters are fitted as ln c = -26.4723 and m = 2.9308. One of the remaining crack growth trajectories is randomly selected as the target prediction. Three data points from that trajectory are randomly chosen to represent the sparse high-fidelity data obtained from field inspection (red solid dots in Fig. 4.12).

The architecture in Fig. 4.3 is used for the multi-fidelity modeling. The prediction results are shown in Fig. 4.12. The blue dash-dotted line is the Paris model calibrated using the 10 trajectory data. Four trajectories from the remaining dataset are randomly chosen and are shown in orange solid lines. The predictions from the MDA-CNN model are shown in green dashed lines. It can be seen that the crack growth curve from the Paris model deviates from the individual actual trajectories. However, with 3 high-fidelity data (shown in red dots) and the low-fidelity Paris' model, accurate predictions of crack growth trajectories can be obtained using the MDA-CNN.

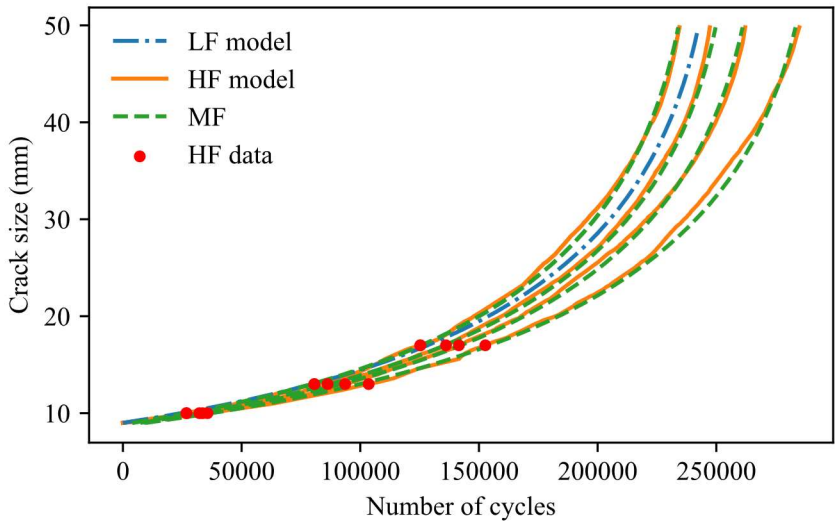
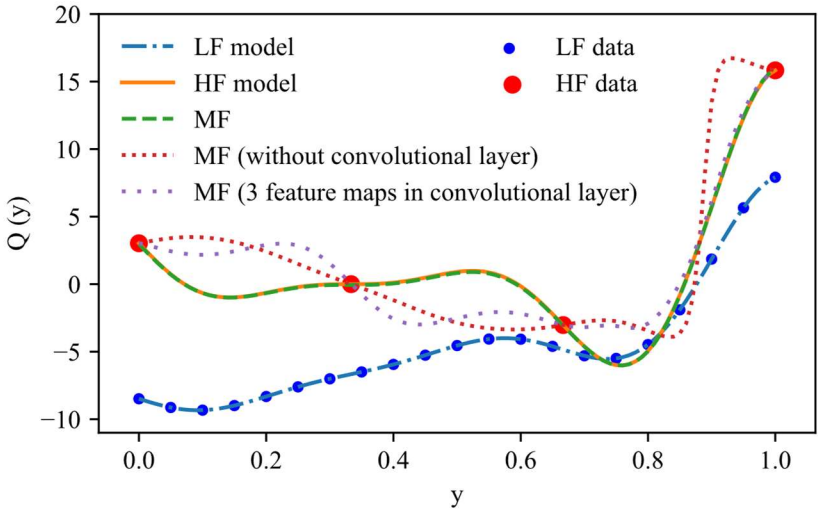


Fig. 4.12 The Results of Fatigue Crack Growth Trajectories Using MDA-CNN.

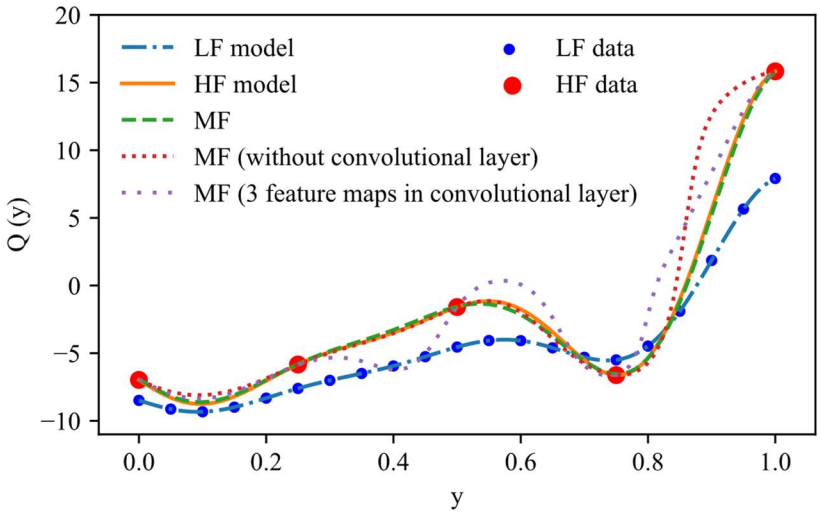
4.2.6 Discussions

Fig. 4.13 (a) and (b) show the comparison between results obtained from a neural network with and without the convolutional layer. Those two figures correspond to continuous function with linear relationship, and continuous function with nonlinear relationship for scenarios (a) and (c) in Table 4.1, respectively. The results with the convolutional layer shown in green dashed lines are obtained using the architecture in Fig. 4.3. The results without the convolutional layer (brown dotted line) are calculated using the architecture in Fig. 4.2. It can be seen that, without the convolutional layer, the predicted results have poor accuracy compared with the ones obtained with the proposed architecture. This is due to the small size of high-fidelity data. With the sparse data, it is insufficient to produce an accurate result if only the low-fidelity data with the same input y as the available high-fidelity data are used for learning the relationship, i.e., the relationship shown in Fig. 4.1 (a). However, by using the proposed architecture with a

convolutional layer, satisfactory results can be obtained. That is achieved through utilizing all low-fidelity data and capturing the relationship between a high-fidelity datum with every low-fidelity data.



(a) Continuous function with the linear relationship.



(b) Continuous function with the nonlinear relationship.

Fig. 4.13 Result Comparison for Illustration of the Effect of Convolutional Layer in MDA-CNN.

The multi-fidelity results shown in green dashed lines in Fig. 4.13 are obtained with 64 feature maps in the convolutional layer. To investigate the effect of the number of feature maps on the predicted results, the neural networks are retrained with 3 feature maps. The results are shown in purple dotted lines. For the investigated examples, the predictions are inaccurate with only 3 feature maps in the convolutional layer. That can be explained as follows. Each feature maps function learns a simple localized feature of the relationship between low- and high-fidelity. Thus, a sufficient number of feature maps are needed for a complete capture of the relationship.

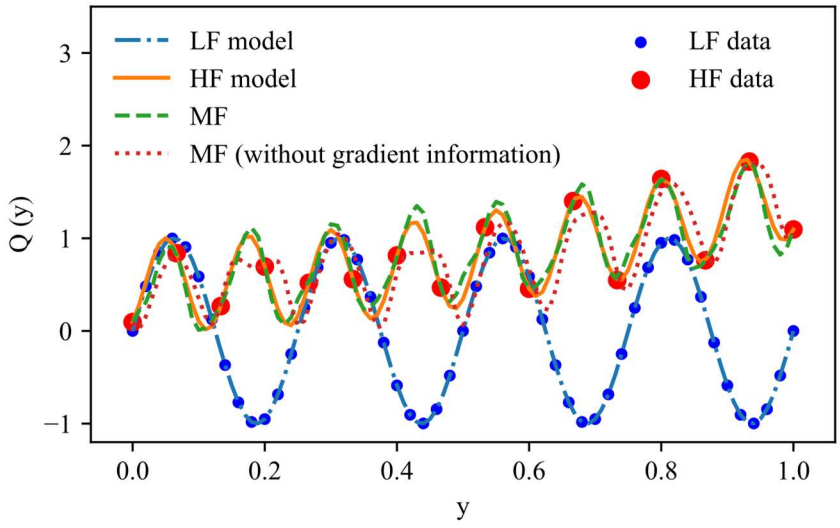
The MF prediction results of Fig. 4.5 (e) and (f) (i.e., Examples of phase-shifted oscillations and different periodicities) are obtained using the input table shown in Fig. 4.4 (c). The first gradient of the low-fidelity model is utilized in the input table. This section discusses the scenarios where the MF modeling is conducted with and without low-fidelity gradient information. The results for the above two examples are shown in Fig. 4.14 (a) and (b), respectively. The green dashed lines are the results obtained by incorporating the first derivative of the low-fidelity model in the input table, and the brown dotted lines are for the results without considering gradient information (i.e., only using the input table in Fig. 4.3). It can be observed that the predictions without low-fidelity gradient information are inaccurate. This can be explained as follows. The high-fidelity model for Eqs. **Error! Reference source not found.**) and **Error! Reference source not found.**) in Table 4.1 can be further expressed as

$$Q_H(y) = y^2 + [Q_L(y)\cdot\cos(\pi/10) + Q_L^{(1)}(y)\cdot\sin(\pi/10)/(8\pi)]^2, \tag{4.21}$$

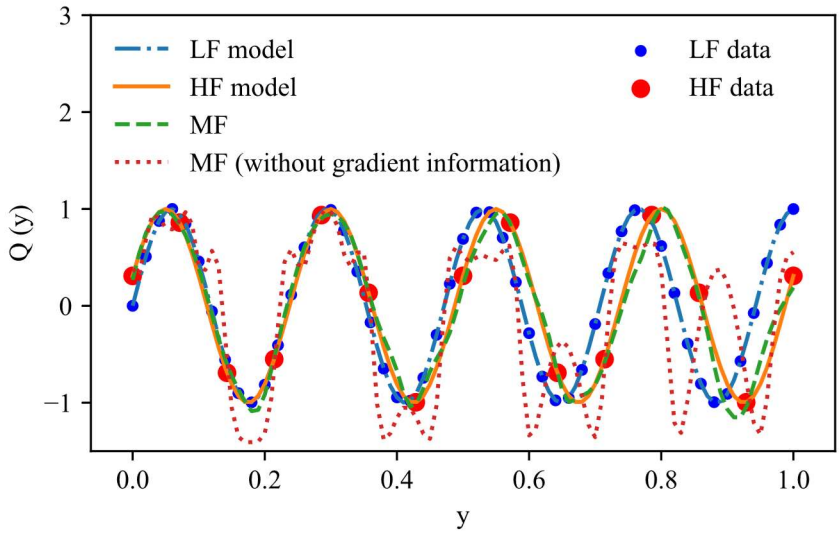
and

$$Q_H(y) = [\cos(by)Q_L(y) - \frac{1}{a}\sin(by)Q_L^{(1)}(y)]\cos(\pi/10) + [\sin(by)Q_L(y) + \frac{1}{a}\cos(by)Q_L^{(1)}(y)]\sin(\pi/10) \quad (4.22)$$

respectively, where $a = 6\sqrt{2}\pi$ and $b = 6\sqrt{2}\pi - 8\pi$. The high-fidelity model is a function of not only the low-fidelity model itself but also its first derivative. If no low-fidelity gradient information is provided for multi-fidelity modeling, the present datasets are insufficient for the neural network model to learn the correct relationship [168]. In Ref. [168], this problem is solved by incorporating $Q_L(y - \tau)$ where τ is the delay and viewing that as an implicit approximation of the first derivative. The selection of optimal value for the time delay τ is critical and problem-dependent [168, 183]. The multi-fidelity modeling fails without an optimal τ. However, by explicitly incorporating the first derivative information of the LF model in the proposed MDA-CNN, the time delay τ can be avoided. Thus, the proposed method can be applied with more flexibility for different problems.



(a) Phase-shifted oscillations.



(b) Different periodicity.

Fig. 4.14 Result Comparison for Input Tables with and Without Low Fidelity Gradient Information.

4.3 Uncertainty Quantification of Multi-fidelity Modeling

The multi-fidelity model in Section 4.2 is a deterministic method. That is, no uncertainty is considered. That model will be further expanded to be able to quantify the uncertainty. Bayesian statistics offer a formalism to understand and quantify the uncertainty associated with deep neural networks predictions. Thus, the multi-fidelity neural network needs to be improved by utilizing the idea from Bayesian neural networks.

4.4 Conclusions

This chapter presents a novel methodology named Multi-fidelity Data Aggregation using Convolutional Neural Networks (MDA-CNN) for multi-fidelity modeling. The MDA-CNN is composed of three components: multi-fidelity data compiling, multi-fidelity

perceptive field and convolution, and a deep neural network for mapping. This framework fully exploits the relationship between low- and high-fidelity data. That is, it aims to capture and utilize the relationship between any high-fidelity data with all available low-fidelity data, instead of just a point-to-point relationship (i.e., a high-fidelity datum with one corresponding low-fidelity datum). That is achieved by incorporating all the low-fidelity data and sliding the local receptive field connected to hidden neurons in the convolutional layer across the entire low-fidelity data. This framework can be easily adapted for the scenarios with multiple low-fidelity models, high-dimensional inputs, incorporating additional low-fidelity information, etc. Those can be achieved by properly designing the input table.

The viability of the MDA-CNN has been demonstrated using extensive numerical examples including the linear and nonlinear relationship between low- and high-fidelity functions, discontinuous functions, oscillation functions with phase shift and different periodicities, and high-dimensional models. We also discuss and compare the results with/without the convolutional layer, and with/without additional low-fidelity information (derivatives). After validation, the MDA-CNN is applied to solve two engineering problems with different types of levels of fidelities, stress prediction with coarse vs. fine mesh in finite element analysis, and fatigue crack growth with simplified physics model vs. experimental data. In both numerical and engineering examples, accurate results can be obtained with the proposed framework.

5 CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

This study works on the prediction and uncertainty quantification of material mechanical properties (static and fatigue). Corresponding to the four objectives described in Chapter 1, the following conclusions can be drawn.

1. Bayesian model averaging method is carried out for multimodality information fusion to estimate the probabilistic vintage pipe strength. The basic theory and practical implementation of BMA are introduced. Each model considered to be averaged is assigned a weight according to its marginal likelihood and further posterior model probability. BMA is implemented through distribution mixing. Occam's window method is used to reduce the number of models considered to improve the feasibility and efficiency of BMA as well as keep the efficacy. The predictive performance of BMA and the single model are evaluated and compared according to the logarithmic scoring rule. Bayesian model averaging has the advantage of improving the predictive ability by considering model uncertainty when the number of data is insufficient to select a single best model.
2. To obtain the probabilistic *S*-*N* curves with sparse data, this work proposes four strategies: hierarchical Bayesian modeling (HBM), Bayesian data augmentation (BDA), posterior information from HBM used as prior information for BDA (HBM+BDA), and augmented data from BDA used by HBM (BDA+HBM). HBM+BDA and BDA+HBM are together called hierarchical Bayesian data augmentation (HBDA). The theories and implementations are illustrated. The four

strategies are validated using aluminum alloy 2524-T3 data, and laminate panel data. After that, the proposed methods are applied to estimate the probabilistic *S-N* curves for the Pearl Harbor Memorial Bridge. HBDA shows a large performance gain, especially when the number of the testing specimens is low (e.g., 3), and HBM+BDA and BDA+HBM have similar performance based on the investigated materials.

3. The Probabilistic Physics-guided Neural Network (PPgNN) proposed provides a flexible and robust tool for probabilistic fatigue S-N estimation. Compared with an explicit regression model, the PPgNN is not restricted to a single function format when fitting with multiple factors and at different stress/strain regimes. It also avoids the non-physical predictions occurring for the machine learning model and produces both accurate and physically consistent results.

   The PPgNN model is also able to be trained using the data set with missing data for more reliable predictions. Experimental fatigue data are collected from extensive literature for additively manufactured Ti-64 subjected to various in-process and post-process heat treatment parameters. The PPgNN model is validated using the experimental data.

   Mixture Density Network (PgMDN) integrates a Mixture Density Network for probabilistic modeling and physics knowledge as regularization. This model has no limitations on the distribution types of data. The physics-guided machine learning is modeled as a constrained optimization problem. Both equality and inequality constraints are considered. A dynamic penalty function algorithm is

proposed for training the NN. With the physics constraints, the needed data size can be reduced.

4. A novel methodology named Multi-fidelity Data Aggregation using Convolutional Neural Networks (MDA-CNN) is proposed for multi-fidelity modeling. This framework fully exploits the relationship between low- and high-fidelity data. That is, it aims to capture and utilize the relationship between any high-fidelity data with all available low-fidelity data, instead of just a point-to-point relationship (i.e., a high-fidelity datum with one corresponding low-fidelity datum). This framework can be easily adapted for the scenarios with multiple low-fidelity models, high-dimensional inputs, incorporating additional low-fidelity information, etc. Those can be achieved by properly designing the input table.

5.2 Future Work

Future work will concentrate on the following aspects:

1. Chapter 2 considers model uncertainty due to variable selection by constructing linear models. Future work may concentrate on the other forms of model uncertainties. BMA may extent to consider model uncertainties from the functional forms. Also, the likelihood for the data is assumed to be Gaussian distribution. The influence of different likelihood distributions can be further studied.
2. The PgMDN presented in Chapter 3 focuses on the methodology illustration and does not address the scalability issue for high-dimensional data analysis. Dimension reduction technique and reduced-order modeling can be coupled with the proposed methodology for uncertainty quantification and needs further study.

In many engineering analyses, multiscale modeling is commonly used and uncertainties from different scales need to be quantified. The proposed methodology needs to be extended to handle multiscale UQ problems. A possible direction is to construct multiple MDNs with correlations between them embedded as constraints. Finally, the current study has an implicit hypothesis that the encoded physics constraints are correct and has no errors. If physics knowledge is not perfect or might be changed due to a change of environments (such as those in the transfer learning problems), automatic adaptation between new data and prior physics constraints is desired.

3. The MDA-CNN proposed in Chapter 4 is a fundamental model that introduces convolutional neural network (CNN) in multi-fidelity (and multi-source) modeling for the first time. Several future research directions are presented based on the current study. First, the MDA-CNN presented in this work uses only one convolutional layer and no pooling layers. This is due to the relatively low dimension of data investigated in this work. For higher-dimensional and more complicated data, more convolutional and pooling layers can be utilized. Second, the local receptive field sliding across the compiled input table helps to learn the relationship between high- and low-fidelity data locally and sequentially. Other manners of moving the local receptive field can be explored for a more effective relationship capturing, for example, noncontinuous sliding.

REFERENCES

[1] D.C. Montgomery, G.C. Runger, N.F. Hubele, Engineering statistics, John Wiley & Sons, 2009.

[2] F.A. Morrison, Uncertainty Analysis for Engineers and Scientists: A Practical Guide, Cambridge University Press, 2020.

[3] F. Canavero, Uncertainty Modeling for Engineering Applications, Springer, 2019.

[4] A. Gelman, J.B. Carlin, H.S. Stern, D.B. Dunson, A. Vehtari, D.B. Rubin, Bayesian data analysis, Chapman and Hall/CRC, 2013.

[5] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, MIT press, 2016.

[6] T. Hastie, R. Tibshirani, J. Friedman, The elements of statistical learning: data mining, inference, and prediction, Springer Science & Business Media, 2009.

[7] K.P. Murphy, Machine learning: a probabilistic perspective, MIT press, 2012.

[8] J. Chen, D. Ersoy, Y. Liu, Probabilistic bulk property estimation using multimodality surface non-destructive measurements for vintage pipes, Structural Safety, 87 (2020) 101995.

[9] J. Chen, S. Liu, W. Zhang, Y. Liu, Uncertainty quantification of fatigue S-N curves with sparse data using hierarchical Bayesian data augmentation, International Journal of Fatigue, 134 (2020) 105511.

[10] M. Mishra, V. Keshavarzzadeh, A. Noshadravan, Reliability-based lifecycle management for corroding pipelines, Structural Safety, 76 (2019) 1-14.

[11] S.A. Timashev, A.V. Bushinskaya, Markov approach to early diagnostics, reliability assessment, residual life and optimal maintenance of pipeline systems, Structural Safety, 56 (2015) 68-79.

[12] S. Dahire, F. Tahir, Y. Jiao, Y. Liu, Bayesian Network inference for probabilistic strength estimation of aging pipeline systems, International Journal of Pressure Vessels and Piping, 162 (2018) 30-39.

[13] Z. Li, S. Schmauder, M. Dong, A simple mechanical model to predict fracture and yield strengths of particulate two-phase materials, Computational Materials Science, 15 (1999) 11-21.

[14] B. Bramfitt, Structure/Property relationships in irons and steels, Metals Handbook Desk Edition, 20 (1998) 153-173.

[15] S.H. Hashemi, Strength–hardness statistical correlation in API X65 steel, Materials Science and Engineering: A, 528 (2011) 1648-1655.

[16] J. Chen, B. Diao, J. He, S. Pang, X. Guan, Equivalent surface defect model for fatigue life prediction of steel reinforcing bars with pitting corrosion, International Journal of Fatigue, 110 (2018) 153-161.

[17] Q. Duan, N. Ajami, X. Gao, S. Sorooshian, Multi-Model Ensemble Hydrologic Prediction Using Bayesian Model Averaging, 30 (2007) 1371-1386.

[18] A.E. Raftery, T. Gneiting, F. Balabdaoui, M. Polakowski, Using Bayesian Model Averaging to Calibrate Forecast Ensembles, 133 (2005) 1155-1174.

[19] Y. Honjo, L. Wen-Tsung, S. Sakajo, Application of Akaike information criterion statistics to geotechnical inverse analysis: the extended Bayesian method, Structural Safety, 14 (1994) 5-29.

[20] K.P. Burnham, D.R. Anderson, Multimodel Inference: Understanding AIC and BIC in Model Selection, 33 (2004) 261-304.

[21] G. Li, J. Shi, Application of Bayesian model averaging in modeling long-term wind speed distributions, Renewable Energy, 35 (2010) 1192-1202.

[22] T.M. Fragoso, W. Bertoli, F. Louzada, Bayesian Model Averaging: A Systematic Review and Conceptual Classification, 86 (2018) 1-28.

[23] J.A. Hoeting, D. Madigan, A.E. Raftery, C.T. Volinsky, Bayesian Model Averaging: A Tutorial, Statistical Science, 14 (1999) 382-401.

[24] P. Gregory, Bayesian Logical Data Analysis for the Physical Sciences: A Comparative Approach with Mathematica® Support, Cambridge University Press, Cambridge, 2005.

[25] D. Madigan, A.E. Raftery, Model Selection and Accounting for Model Uncertainty in Graphical Models Using Occam's Window, Journal of the American Statistical Association, 89 (1994) 1535-1546.

[26] G. Claeskens, N.L.J.C.B. Hjort, Model selection and model averaging, Cambridge University Press, Cambridge, 2008.

[27] R. van Haasteren, Marginal Likelihood Calculation with MCMC Methods, in: Gravitational Wave Detection and Data Analysis for Pulsar Timing Arrays, Springer, Berlin, Heidelberg, 2014, pp. 99-120.

[28] X. Guan, R. Jha, Y. Liu, Model selection, updating, and averaging for probabilistic fatigue damage prognosis, Structural Safety, 33 (2011) 242-249.

[29] D. Madigan, A.E. Raftery, J.C. York, J.M. Bradshaw, R.G. Almond, Strategies for Graphical Model Selection, in, Springer New York, New York, NY, 1994, pp. 91-100.

[30] E.P.B. George, Sampling and Bayes' Inference in Scientific Modelling and Robustness, Journal of the Royal Statistical Society. Series A (General), 143 (1980) 383-430.

[31] G. Nguefack-Tsague, W. Zucchini, A Mixture-Based Bayesian Model Averaging Method, 2016.

[32] A. Raftery, D. Madigan, C. T. Volinsky, Accounting for Model Uncertainty in Survival Analysis Improves Predictive Performance, 2000.

[33] J. Piironen, A.J.S. Vehtari, Computing, Comparison of Bayesian predictive methods for model selection, 27 (2017) 711-735.

[34] J. Ji, D.J. Robert, C. Zhang, D. Zhang, J. Kodikara, Probabilistic physical modelling of corroded cast iron pipes for lifetime prediction, Structural Safety, 64 (2017) 62-75.

[35] J. He, X. Guan, R. Jha, Improve the Accuracy of Asymptotic Approximation in Reliability Problems Involving Multimodal Distributions, IEEE Transactions on Reliability, 65 (2016) 1724-1736.

[36] S. Chib, E. Greenberg, Understanding the Metropolis-Hastings Algorithm, The American Statistician, 49 (1995) 327-335.

[37] M.A. Newton, A.E. Raftery, Approximate Bayesian Inference with the Weighted Likelihood Bootstrap, Journal of the Royal Statistical Society. Series B (Methodological), 56 (1994) 3-48.

[38] P.D. Hoff, A first course in Bayesian statistical methods, Springer, 2009.

[39] T. Havranek, Z. Irsova, O. Zeynalova, Tuition Fees and University Enrolment: A Meta-Regression Analysis, Oxf. Bull. Econ. Stat., 80 (2018) 1145-1184.

[40] H. Jeffreys, Theory of probability, Clarendon, in, Oxford, 1961.

[41] J. Kuha, AIC and BIC:Comparisons of Assumptions and Performance, 33 (2004) 188-229.

[42] G. Schwarz, Estimating the Dimension of a Model, The Annals of Statistics, 6 (1978) 461-464.

[43] J. Evans, J. Sullivan, Approximating model probabilities in Bayesian information criterion and decision-theoretic approaches to model selection in phylogenetics, Mol Biol Evol, 28 (2011) 343-349.

[44] F. Pascual, W. Meeker, Estimating Fatigue Curves with the Random Fatigue-Limit Model, Technometrics, 41 (1999) 277-290.

[45] F. Liu, S. Zhou, C. Xia, D. Zeng, T. Shi, Optimization of fatigue life distribution model and establishment of probabilistic S–N curves for a 165ksi grade super high strength drill pipe steel, Journal of Petroleum Science and Engineering, 145 (2016) 527-532.

[46] Y. Liu, S. Mahadevan, Probabilistic fatigue life prediction using an equivalent initial flaw size distribution, International Journal of Fatigue, 31 (2009) 476-487.

[47] J. Wu, B. Diao, J. Xu, R. Zhang, W. Zhang, Effects of the reinforcement ratio and chloride corrosion on the fatigue behavior of RC beams, International Journal of Fatigue, 131 (2020) 105299.

[48] J. Wu, R. Zhang, B. Diao, W. Zhang, J. Xu, Effects of pre-fatigue damage on high-cycle fatigue behavior and chloride permeability of RC beams, International Journal of Fatigue, 122 (2019) 9-18.

[49] J. He, X. Guan, T. Peng, Y. Liu, A. Saxena, J. Celaya, K. Goebel, A multi-feature integration method for fatigue crack detection and crack length estimation in riveted lap joints using Lamb waves, Smart Materials and Structures, 22 (2013) 105007.

[50] X.-W. Liu, D.-G. Lu, P.C.J. Hoogenboom, Hierarchical Bayesian fatigue data analysis, International Journal of Fatigue, 100 (2017) 418-428.

[51] X. Guan, J. He, Life time extension of turbine rotating components under risk constraints: A state-of-the-art review and case study, International Journal of Fatigue, 129 (2019) 104799.

[52] H. Wei, Y. Liu, A critical plane-energy model for multiaxial fatigue life prediction, Fatigue & Fracture of Engineering Materials & Structures, 40 (2017) 1973-1983.

[53] Z. Guo, Y. Ma, L. Wang, J. Zhang, H. E, Corrosion fatigue crack propagation mechanism of high strength steel bar in various environments, Journal of Materials in Civil Engineering, 32 (2020) 04020115.

[54] L. Xie, J. Liu, N. Wu, W. Qian, Backwards statistical inference method for P–S–N curve fitting with small-sample experiment data, International Journal of Fatigue, 63 (2014) 62-67.

[55] Y. Ma, Z. Guo, L. Wang, J. Zhang, Probabilistic life prediction for reinforced concrete structures subjected to seasonal corrosion-fatigue damage, Journal of Structural Engineering, 146 (2020) 04020117.

[56] J.A. Collins, Failure of materials in mechanical design: analysis, prediction, prevention, John Wiley & Sons, 1993.

[57] Y. Liu, S. Mahadevan, Stochastic fatigue damage modeling under variable amplitude loading, International Journal of Fatigue, 29 (2007) 1149-1161.

[58] X. Tan, Prevention, P–S–N Curve Fitting Method Based on Sample Aggregation Principle, Journal of Failure Analysis, 19 (2019) 270-278.

[59] J. Gao, Z. An, B. Liu, A new method for obtaining P-S-N curves under the condition of small sample, 231 (2017) 130-137.

[60] X.-W. Liu, D.-G. Lu, Survival analysis of fatigue data: Application of generalized linear models and hierarchical Bayesian model, International Journal of Fatigue, 117 (2018) 39-46.

[61] A. Gelman, J. Hill, Data analysis using regression and multilevel/hierarchical models, Cambridge university press, 2006.

[62] T.D. Pigott, evaluation, A review of methods for missing data, Educational Research and Evaluation, 7 (2001) 353-383.

[63] X. Zhou, J.P. Reiter, A note on Bayesian inference after multiple imputation, The American Statistician, 64 (2010) 159-163.

[64] B. Walczak, D.L. Massart, Dealing with missing data: Part I, Chemometrics and Intelligent Laboratory Systems, 58 (2001) 15-27.

[65] B. Walczak, D.L. Massart, Dealing with missing data: Part II, Chemometrics and Intelligent Laboratory Systems, 58 (2001) 29-42.

[66] J.I. Park, S.J. Bae, Direct prediction methods on lifetime distribution of organic light-emitting diodes from accelerated degradation tests, IEEE Transactions on Reliability, 59 (2010) 74-90.

[67] C.K. Wikle, Hierarchcal Baysian models for predicting the spread of ecological processes, 84 (2003) 1382-1394.

[68] O. Basquin, The exponential law of endurance tests, Proc Am Soc Test Mater, 10 (1910) 625-630.

[69] D.G. Clayton, Generalized linear mixed models, Markov chain Monte Carlo in practice, 1 (1996) 275-302.

[70] W.R. Gilks, S. Richardson, D.J. Spiegelhalter, Introducing markov chain monte carlo, Markov chain Monte Carlo in practice, 1 (1996) 19.

[71] W.R. Gilks, Full conditional distributions, Markov chain Monte Carlo in practice, (1996) 75-88.

[72] Y. Wang, Y. Liu, Bayesian entropy network for fusion of different types of information, Reliability Engineering & System Safety, 195 (2020) 106747.

[73] J.L. Schafer, Analysis of incomplete multivariate data, Chapman and Hall/CRC, 1997.

[74] M.A. Tanner, W.H. Wong, The calculation of posterior distributions by data augmentation, Journal of the American Statistical Association, 82 (1987) 528-540.

[75] R. Lall, How multiple imputation makes a difference, Political Analysis, 24 414-433.

[76] D. Ni, J.D. Leonard, A. Guin, C. Feng, Multiple imputation scheme for overcoming the missing values and variability issues in ITS data, 131 (2005) 931-938.

[77] D.B. Rubin, Multiple imputation for nonresponse in surveys, John Wiley & Sons, 2004.

[78] J.W. Graham, A.E. Olchowski, T.D. Gilreath, How Many Imputations are Really Needed? Some Practical Clarifications of Multiple Imputation Theory, Prevention Science, 8 (2007) 206-213.

[79] A.R.T. Donders, G.J.M.G. van der Heijden, T. Stijnen, K.G.M. Moons, Review: A gentle introduction to imputation of missing values, Journal of Clinical Epidemiology, 59 (2006) 1087-1091.

[80] J.L. Schafer, Multiple imputation: a primer, 8 (1999) 3-15.

[81] P.H. Rezvan, K.J. Lee, J.A. Simpson, The rise of multiple imputation: a review of the reporting and implementation of the method in medical research, BMC medical research methodology, 15 (2015) 30.

[82] T. Shimokawa, Y. Hamaguchi, Statistical evaluation of fatigue life and fatigue strength in circular- hole notched specimens of a carbon eight-harness-satin/epoxy laminate, Statistical Research on Fatigue, (1987) 159-176.

[83] J. Chen, Y. Liu, Probabilistic physics-guided machine learning for fatigue data analysis, Expert Systems with Applications, (2020) 114316.

[84] J. Chen, Y. Liu, Fatigue property prediction of additively manufactured Ti-6Al-4V using probabilistic physics-guided learning, Additive Manufacturing, 39 (2021) 101876.

[85] J. Chen, Y. Liu, Uncertainty quantification of fatigue properties with sparse data using hierarchical Bayesian model, in: Proc. AIAA Scitech 2020 Forum, American Institute of Aeronautics and Astronautics, 2020.

[86] Y. Liu, S. Mahadevan, Efficient Methods for Time-Dependent Fatigue Reliability Analysis, AIAA Journal, 47 (2009) 494-504.

[87] Y.-H. Kim, J.-H. Song, J.-H. Park, An expert system for fatigue life prediction under variable loading, Expert Systems with Applications, 36 (2009) 4996-5008.

[88] J. Chen, A. Imanian, H. Wei, N. Iyyer, Y. Liu, Piecewise stochastic rainflow counting for probabilistic linear and nonlinear damage accumulation considering loading and material uncertainties, International Journal of Fatigue, 140 (2020) 105842.

[89] Y. Liu, L. Liu, B. Stratman, S. Mahadevan, Multiaxial fatigue reliability analysis of railroad wheels, Reliability Engineering & System Safety, 93 (2008) 456-467.

[90] X. Liu, F.-Z. Xuan, J. Si, S.-T. Tu, Expert system for remnant life prediction of defected components under fatigue and creep–fatigue loadings, Expert Systems with Applications, 34 (2008) 222-230.

[91] C.E. Stromeyer, W.E. Dalby, The determination of fatigue limits under alternating stress conditions, Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character, 90 (1914) 411-425.

[92] L. D'Angelo, A. Nussbaumer, Estimation of fatigue S-N curves of welded joints using advanced probabilistic approach, International Journal of Fatigue, 97 (2017) 98-113.

[93] D. Leonetti, J. Maljaars, H.H. Snijder, Fitting fatigue test data with a novel S-N curve using frequentist and Bayesian inference, International Journal of Fatigue, 105 (2017) 128-143.

[94] H. Yaghobi, H. Rajabi Mashhadi, K. Ansari, Artificial neural network approach for locating internal faults in salient-pole synchronous generator, Expert Systems with Applications, 38 (2011) 13328-13341.

[95] T.T. Pleune, O.K. Chopra, Artificial neural networks and the effects of loading conditions on fatigue life of carbon and low-alloy steels, in, Oak Ridge Inst. for Science and Education, TN (United States), 1996.

[96] P. Artymiak, L. Bukowski, J. Feliks, S. Narberhaus, H. Zenner, Determination of S–N curves with the application of artificial neural networks, Fatigue & Fracture of Engineering Materials & Structures, 22 (1999) 723-728.

[97] A.P. Vassilopoulos, E.F. Georgopoulos, V. Dionysopoulos, Artificial neural networks in spectrum fatigue life prediction of composite materials, International Journal of Fatigue, 29 (2007) 20-29.

[98] J.C. Figueira Pujol, J.M. Andrade Pinto, A neural network approach to fatigue life prediction, International Journal of Fatigue, 33 (2011) 313-322.

[99] J.F. Barbosa, J.A.F.O. Correia, R.C.S.F. Júnior, A.M.P.D. Jesus, Fatigue life prediction of metallic materials considering mean stress effects by means of an artificial neural network, International Journal of Fatigue, 135 (2020) 105527.

[100] M.A. Herzog, T. Marwala, P.S. Heyns, Machine and component residual life estimation through the application of neural networks, Reliability Engineering & System Safety, 94 (2009) 479-489.

[101] X. Jia, J. Willard, A. Karpatne, J.S. Read, J.A. Zwart, M. Steinbach, V. Kumar, Physics-Guided Machine Learning for Scientific Discovery: An Application in Simulating Lake Temperature Profiles, arXiv preprint arXiv:2001.11086, (2020).

[102] H. Xu, B. Yu, Automatic thesaurus construction for spam filtering using revised back propagation neural network, Expert Systems with Applications, 37 (2010) 18-23.

[103] N. Ketkar, Deep Learning with Python, Springer, 2017.

[104] S.A. Kalogirou, Solar energy engineering: processes and systems, Academic Press, 2013.

[105] S. Zychlinski, Predicting probability distributions using neural networks, in, Taboola Engineering, 2018.

[106] J. He, J. Chen, X. Guan, Lifetime distribution selection for complete and censored multi-level testing data and its influence on probability of failure estimates, Structural and Multidisciplinary Optimization, (2020) 1-17.

[107] S. Kumbhar, R. Tayade, A case study on effect of mean stress on fatigue life, International Journal of Engineering Development and Research, 2 (2014) 304-308.

[108] C.-L. Shen, The statistical analysis of fatigue data, in, ProQuest Dissertations Publishing, 1994.

[109] T. Zhao, Y. Jiang, Fatigue of 7075-T651 aluminum alloy, International Journal of Fatigue, 30 (2008) 834-849.

[110] F. Pascual, The random fatigue-limit model in multi-factor experiments, Journal of Statistical Computation and Simulation, 73 (2003) 733-752.

[111] M. Benedetti, V. Fontanari, M. Bandini, F. Zanini, S. Carmignato, Low- and high-cycle fatigue resistance of Ti-6Al-4V ELI additively manufactured via selective laser melting: Mean stress and defect sensitivity, International Journal of Fatigue, 107 (2018) 96-109.

[112] S.P. Narra, Z. Wu, R. Patel, J. Capone, M. Paliwal, J. Beuth, A. Rollett, Use of Non-Spherical Hydride-Dehydride (HDH) Powder in Powder Bed Fusion Additive Manufacturing, Additive Manufacturing, 34 (2020) 101188.

[113] P. Kumar, U. Ramamurty, High cycle fatigue in selective laser melted Ti-6Al-4V, Acta Materialia, 194 (2020) 305-320.

[114] R. Biswal, A.K. Syed, X. Zhang, Assessment of the effect of isolated porosity defects on the fatigue performance of additive manufactured titanium alloy, Additive Manufacturing, 23 (2018) 433-442.

[115] H. Gong, K. Rafi, H. Gu, G.D. Janaki Ram, T. Starr, B. Stucker, Influence of defects on mechanical properties of Ti–6Al–4V components produced by selective laser melting and electron beam melting, Materials & Design, 86 (2015) 545-554.

[116] E. Wycisk, C. Emmelmann, S. Siddique, F. Walther, High cycle fatigue (HCF) performance of Ti-6Al-4V alloy processed by selective laser melting, Advanced materials research, 816 (2013) 134-139.

[117] J.W. Pegues, S. Shao, N. Shamsaei, N. Sanaei, A. Fatemi, D.H. Warner, P. Li, N. Phan, Fatigue of additive manufactured Ti-6Al-4V, Part I: The effects of powder feedstock, manufacturing, and post-process conditions on the resulting microstructure and defects, International Journal of Fatigue, 132 (2020) 105358.

[118] H.R. Sandgren, Y. Zhai, D.A. Lados, P.A. Shade, J.C. Schuren, M.A. Groeber, P. Kenesei, A.G. Gavras, Characterization of fatigue crack growth behavior in LENS fabricated Ti-6Al-4V using high-energy synchrotron x-ray microtomography, Additive Manufacturing, 12 (2016) 132-141.

[119] J. Günther, D. Krewerth, T. Lippmann, S. Leuders, T. Tröster, A. Weidner, H. Biermann, T. Niendorf, Fatigue life of additively manufactured Ti–6Al–4V in the very high cycle fatigue regime, International Journal of Fatigue, 94 (2017) 236-245.

[120] S. Liu, Y.C. Shin, Additive manufacturing of Ti6Al4V alloy: A review, Materials & Design, 164 (2019) 107552.

[121] P. Li, D.H. Warner, A. Fatemi, N. Phan, Critical assessment of the fatigue performance of additively manufactured Ti–6Al–4V and perspective for future research, International Journal of Fatigue, 85 (2016) 130-143.

[122] S.M.J. Razavi, B. Van Hooreweder, F. Berto, Effect of build thickness and geometry on quasi-static and fatigue behavior of Ti-6Al-4V produced by Electron Beam Melting, Additive Manufacturing, 36 (2020) 101426.

[123] H. Shipley, D. McDonnell, M. Culleton, R. Coull, R. Lupoi, G. O'Donnell, D. Trimble, Optimisation of process parameters to address fundamental challenges during selective laser melting of Ti-6Al-4V: A review, International Journal of Machine Tools and Manufacture, 128 (2018) 1-20.

[124] S. Ghouse, S. Babu, K. Nai, P.A. Hooper, J.R.T. Jeffers, The influence of laser parameters, scanning strategies and material on the fatigue strength of a stochastic porous structure, Additive Manufacturing, 22 (2018) 290-301.

[125] M. Fousová, D. Vojtěch, K. Doubrava, M. Daniel, C.-F. Lin, Influence of inherent surface and internal defects on mechanical properties of additively manufactured Ti6Al4V alloy: Comparison between selective laser melting and electron beam melting, Materials, 11 (2018) 537.

[126] H.K. Rafi, T.L. Starr, B.E. Stucker, A comparison of the tensile, fatigue, and fracture behavior of Ti–6Al–4V and 15-5 PH stainless steel parts made by selective laser melting, The International Journal of Advanced Manufacturing Technology, 69 (2013) 1299-1309.

[127] A. Fatemi, R. Molaei, S. Sharifimehr, N. Phan, N. Shamsaei, Multiaxial fatigue behavior of wrought and additive manufactured Ti-6Al-4V including surface finish effect, International Journal of Fatigue, 100 (2017) 347-366.

[128] V. Chastand, P. Quaegebeur, W. Maia, E. Charkaluk, Comparative study of fatigue properties of Ti-6Al-4V specimens built by electron beam melting (EBM) and selective laser melting (SLM), Materials Characterization, 143 (2018) 76-81.

[129] S.S. Razvi, S. Feng, A. Narayanan, Y.-T.T. Lee, P. Witherell, A review of machine learning applications in additive manufacturing, in: ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers Digital Collection, 2019.

[130] X. Qi, G. Chen, Y. Li, X. Cheng, C. Li, Applying Neural-Network-Based Machine Learning to Additive Manufacturing: Current Applications, Challenges, and Future Perspectives, Engineering, 5 (2019) 721-729.

[131] Z. Hu, S. Mahadevan, Uncertainty quantification and management in additive manufacturing: current status, needs, and opportunities, The International Journal of Advanced Manufacturing Technology, 93 (2017) 2855-2874.

[132] Z. Hu, S. Mahadevan, Uncertainty quantification in prediction of material properties during additive manufacturing, Scripta Materialia, 135 (2017) 135-140.

[133] P. Li, D.H. Warner, N. Phan, Predicting the fatigue performance of an additively manufactured Ti-6Al-4V component from witness coupon behavior, Additive Manufacturing, 35 (2020) 101230.

[134] N. Lopes, B. Ribeiro, Handling missing values via a neural selective input model, Neural Network World, 22 (2012) 357.

[135] P.E. Carrion, A. Soltani-Tehrani, S.M. Thompson, N. Shamsaei, Effect of Powder Degradation on the Fatigue Behavior of Additively Manufactured As-Built Ti-6Al-4V, Solid freeform fabrication, (2018).

[136] P.E. Carrion, A. Soltani-Tehrani, N. Phan, N. Shamsaei, Powder Recycling Effects on the Tensile and Fatigue Behavior of Additively Manufactured Ti-6Al-4V Parts, JOM, 71 (2019) 963-973.

[137] K.D. Rekedal, Investigation of the high-cycle fatigue life of selective laser melted and hot isostatically pressed Ti-6Al-4V, in, AIR FORCE INSTITUTE OF TECHNOLOGY WRIGHT-PATTERSON AFB OH GRADUATE SCHOOL OF …, 2015.

[138] G. Nicoletto, S. Maisano, M. Antolotti, F. Dall'Aglio, Influence of post fabrication heat treatments on the fatigue behavior of Ti-6Al-4V produced by selective laser melting, Procedia Structural Integrity, 7 (2017) 133-140.

[139] E. Wycisk, A. Solbach, S. Siddique, D. Herzog, F. Walther, C. Emmelmann, Effects of Defects in Laser Additive Manufactured Ti-6Al-4V on Fatigue Properties, Physics Procedia, 56 (2014) 371-378.

[140] W. Sun, Y.e. Ma, W. Huang, W. Zhang, X. Qian, Effects of build direction on tensile and fatigue performance of selective laser melting Ti6Al4V titanium alloy, International Journal of Fatigue, 130 (2020) 105260.

[141] X. Guan, J. He, R. Jha, Y. Liu, An efficient analytical Bayesian method for reliability and system response updating based on Laplace and inverse first-order reliability computations, Reliability Engineering & System Safety, 97 (2012) 1-13.

[142] A. Urbina, S. Mahadevan, T.L. Paez, A Bayes network approach to uncertainty quantification in hierarchically developed computational models, International Journal for Uncertainty Quantification, 2 (2012).

[143] Y. Wang, Y. Pang, O. Chen, H.N. Iyer, P. Dutta, P.K. Menon, Y. Liu, Uncertainty quantification and reduction in aircraft trajectory prediction using Bayesian-Entropy information fusion, Reliability Engineering & System Safety, 212 (2021) 107650.

[144] A. Urbina, S. Mahadevan, T.L. Paez, Quantification of margins and uncertainties of complex systems in the presence of aleatoric and epistemic uncertainty, Reliability Engineering & System Safety, 96 (2011) 1114-1125.

[145] Y. Yu, H. Yao, Y. Liu, Structural dynamics simulation using a novel physics-guided machine learning method, Engineering Applications of Artificial Intelligence, 96 (2020) 103947.

[146] J. Chen, Y. Liu, Multimodality data fusion for probabilistic strength estimation of aging materials using Bayesian networks, in: Proc. AIAA Scitech 2020 Forum, 2020.

[147] C.M. Bishop, Mixture density networks, in: Neural Computing Research Group Report NCRG/4288, Aston University, 1994.

[148] C.N. Davis, T.D. Hollingsworth, Q. Caudron, M.A. Irvine, The use of mixture density networks in the emulation of complex epidemiological individual-based models, PLoS Comput Biol, 16 (2020) e1006869-e1006869.

[149] A.E. Lovell, A.T. Mohan, P. Talou, Quantifying uncertainties on fission fragment mass yields with mixture density networks, Journal of Physics G: Nuclear and Particle Physics, 47 (2020) 114001.

[150] Z. Yang, D. Jha, A. Paul, W.-k. Liao, A. Choudhary, A. Agrawal, A General Framework Combining Generative Adversarial Networks and Mixture Density Networks for Inverse Modeling in Microstructural Materials Design, arXiv preprint arXiv:2101.10553, (2021).

[151] Z. Men, E. Yee, F.-S. Lien, D. Wen, Y. Chen, Short-term wind speed and power forecasting using an ensemble of mixture density neural networks, Renewable Energy, 87 (2016) 203-211.

[152] Y. Zhao, R. Yang, G. Chevalier, R.C. Shah, R. Romijnders, Applying deep bidirectional LSTM and mixture density network for basketball trajectory prediction, Optik, 158 (2018) 266-272.

[153] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational Physics, 378 (2019) 686-707.

[154] D. Zhang, L. Lu, L. Guo, G.E. Karniadakis, Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems, Journal of Computational Physics, 397 (2019) 108850.

[155] L. Yang, X. Meng, G.E. Karniadakis, B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data, Journal of Computational Physics, 425 (2021) 109913.

[156] J. Hron, A. Matthews, Z. Ghahramani, Variational bayesian dropout: pitfalls and fixes, in: International Conference on Machine Learning, PMLR, 2018, pp. 2019-2028.

[157] G. Villarrubia, J.F. De Paz, P. Chamoso, F.D. la Prieta, Artificial neural networks used in optimization problems, Neurocomputing, 272 (2018) 10-16.

[158] A.G. Baydin, B.A. Pearlmutter, A.A. Radul, J.M. Siskind, Automatic differentiation in machine learning: a survey, Journal of machine learning research, 18 (2018).

[159] S.S. Rao, Engineering optimization: theory and practice, John Wiley & Sons, 2019.

[160] Ö. Yeniay, Penalty function methods for constrained optimization with genetic algorithms, Mathematical and computational Applications, 10 (2005) 45-56.

[161] H. Wei, J. Chen, P. Carrion, A. Imanian, N. Shamsaei, N. Iyyer, Y.J.M.W.C. Liu, Multiaxial high-cycle fatigue modelling for random loading, 300 (2019) 12005.

[162] H. Wei, P. Carrion, J. Chen, A. Imanian, N. Shamsaei, N. Iyyer, Y. Liu, Multiaxial high-cycle fatigue life prediction under random spectrum loadings, International Journal of Fatigue, 134 (2020) 105462.

[163] B. Peherstorfer, K. Willcox, M. Gunzburger, Survey of multifidelity methods in uncertainty propagation, inference, and optimization, Siam Review, 60 (2018) 550-591.

[164] L. Bonfiglio, P. Perdikaris, S. Brizzolara, G.E. Karniadakis, Multi-fidelity optimization of super-cavitating hydrofoils, Computer Methods in Applied Mechanics and Engineering, 332 (2018) 63-85.

[165] X. Zhang, F. Xie, T. Ji, Z. Zhu, Y. Zheng, Multi-fidelity deep neural network surrogate model for aerodynamic shape optimization, Computer Methods in Applied Mechanics and Engineering, 373 (2021) 113485.

[166] M.G. Fernández-Godino, C. Park, N.-H. Kim, R.T. Haftka, Review of multi-fidelity models, arXiv preprint arXiv:1609.07196, (2016).

[167] D. Liu, Y. Wang, Multi-Fidelity Physics-Constrained Neural Network and Its Application in Materials Modeling, Journal of Mechanical Design, 141 (2019).

[168] X. Meng, G.E. Karniadakis, A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems, Journal of Computational Physics, 401 (2020) 109020.

[169] S. Saha, Z. Gan, L. Cheng, J. Gao, O.L. Kafka, X. Xie, H. Li, M. Tajdari, H.A. Kim, W.K. Liu, Hierarchical Deep Learning Neural Network (HiDeNN): An artificial intelligence (AI) framework for computational science and engineering, Computer Methods in Applied Mechanics and Engineering, 373 (2021) 113452.

[170] X. Meng, Z. Li, D. Zhang, G.E. Karniadakis, PPINN: Parareal physics-informed neural network for time-dependent PDEs, Computer Methods in Applied Mechanics and Engineering, 370 (2020) 113250.

[171] R.C. Aydin, F.A. Braeu, C.J. Cyron, General Multi-Fidelity Framework for Training Artificial Neural Networks With Computational Models, Frontiers in Materials, 6 (2019).

[172] M. Motamed, A multi-fidelity neural network surrogate sampling method for uncertainty quantification, International Journal for Uncertainty Quantification, 10 (2020).

[173] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770-778.

[174] D. Lazar, Building a ResNet in Keras, in, Towards Data Science, 2020.

[175] M.A. Nielsen, Neural Networks and Deep Learning, Determination Press, 2015.

[176] S. Lee, F. Dietrich, G.E. Karniadakis, I.G. Kevrekidis, Linking Gaussian process regression with data-driven manifold embeddings for nonlinear data fusion, Interface focus, 9 (2019) 20180083.

[177] J.W. Judy, Microelectromechanical systems (MEMS): fabrication, design and applications, Smart materials and Structures, 10 (2001) 1115.

[178] Y. Gao, Y. Jiao, Y. Liu, Ultra-efficient reconstruction of 3D microstructure and distribution of properties of random heterogeneous materials containing multiple phases, Acta Materialia, 204 (2021) 116526.

[179] J. Yang, J. He, X. Guan, D. Wang, H. Chen, W. Zhang, Y. Liu, A probabilistic crack size quantification method using in-situ Lamb wave test and Bayesian updating, Mechanical Systems and Signal Processing, 78 (2016) 118-133.

[180] D.A. Virkler, B.M. Hillberry, P.K. Goel, The statistical nature of fatigue crack propagation, Journal of Engineering Materials and Technology, 101 (1979) 148-153.

[181] S.Z. Feng, X. Han, Z.J. Ma, G. Królczyk, Z.X. Li, Data-driven algorithm for real-time fatigue life prediction of structures with stochastic parameters, Computer Methods in Applied Mechanics and Engineering, 372 (2020) 113373.

[182] P. Paris, F. Erdogan, A critical analysis of crack propagation laws, (1963).

[183] R. Hegger, H. Kantz, T. Schreiber, Practical implementation of nonlinear time series methods: The TISEAN package, Chaos, 9 (1999) 413-435.