January 2019

Deep Learning Based Reliability Models For High Dimensional Data

Mohammad Aminisharifabad
Wayne State University, mohamad00014@gmail.com

DEEP LEARNING BASED RELIABILITY MODELS FOR HIGH DIMENSIONAL DATA

by

MOHAMMAD AMINISHARIFABAD

DISSERTATION

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

2019

MAJOR: INDUSTRIAL ENGEERING

Approved By:

_____

Advisor Date

_____

_____

_____

DEDICATION

To my beloved wife and my family for their love and support.

ACKNOWLEDGEMENT

An exclusive appreciation is extended to Dr. Qingyu Yang for providing me the opportunity to explore areas such as spatial analysis, deep learning and machine learning with application in reliability. During my PhD study, he provided me valuable advices and technical feedback. Also, I would like to acknowledge my advisory committee -- Dr. Leslie Monplaisir, Dr. Evrin Dalkiran, and Dr. Xin Wu.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

1.1 Background and Motivation

Reliability estimation of products/materials is an important issue in current competitive markets. Reliability has application in various industries including agriculture [1], healthcare [2] and material science [3]. The goal of reliability analysis to predict failure time, reduce maintenance cost and operational cost. Hence, accurate reliability estimation is highly important. Although the proposed methodologies of the dissertation can be applied in many applications, we focus on reliability application in material science and steel.

In recent years, advanced high strength steel (AHSS) has received increasing attention in industries due to its high performance such as high strength, low weight, and increased safety. In particular, high strength Dual-Phase (DP) steel is the most widely used AHSS in the automotive industry [4]. DP steel consists of two phases, i.e. martensite and ferrite, where a phase is a type of particle with distinct chemical or physical properties. shows the microstructure of a high strength dual-phase steel sample obtained by an $1000X$ optical microscope, in which black and white pixels refer to martensite and ferrite, respectively.

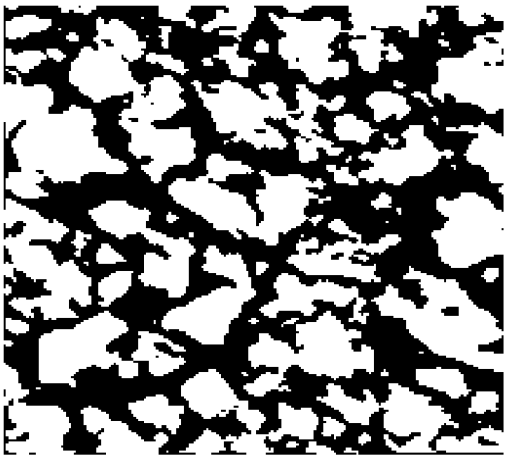

Figure 1. Microstructure sample

In material science, research has shown that steel's microstructure has a strong influence on the mechanical properties of the steel, such as strength, ductility, hardness, toughness, and wear resistance [5, 6]. Furthermore, the microstructure effects failure time of the corresponding material.

To improve the reliability prediction of DP steel, the microstructure of the steel, which is termed as covariates, have to take into account. The goal of this dissertation is to model the covariates and develop methodologies to model effect of the covariates on failure time.

1.2 Related literature

Covariates have topological complex and high dimensional structures. Moreover, the covariates have spatial properties. Modeling of spatial covariates has been studied by researchers in the literature. Paul [7] proposed a simple model which assumes that one phase is uniformly distributed spatially in another phase. Another common model leverages two-point correlation functions to model the material microstructure, in which the two-point correlation functions are defined as the probability that two pixels in the image share the same phase given the relative displacement of the two pixels [8-11]. This model captures the spatial properties of the microstructure using functional data, but the model parameters can be high dimensional with an infinite number of parameters. Feng et al [12] applied a Gaussian random field (GRF) to reconstruct two-phase composite materials with random morphology and model the binary image by translating a GRF to a binary field using a fixed threshold value. Huffer and Wu [13] showed that the method is stationary up to the second order, i.e. the mean and the autocorrelation matrix, in binary images. However, determining the threshold value is difficult in real world case studies.

The autologistic regression model [14] was developed to study the spatial binary data, and the model has been applied in multiple disciplines including ecology [15], agriculture [16] and image analysis [17]. An autologistic regression model assumes the probability that a site belongs to 0 or 1 only depends on its neighbors where neighbors are defined as a collection of image pixels around this site/pixel. Cross and Jain [18] showed that the autologistic regression model is well suited for the binary image that is relevant to the material microstructure modeling problem. Recently Zhang and Yang [19] proposed a model based on the autologistic regression model to capture the microstructure variation of multiple samples for the two phase materials. However, the random effect autologistic model selects arbitrary neighboring order to model multiple microstructure samples which may lead to underfitting or overfitting if low or high order of neighboring are selected, respectively. Moreover, estimation of the model parameters in random effect autologistic becomes challenging as the number of random and fixed components increases.

The existing statistical model are not efficient to model complex spatial data. Specifically, existing models consider that each point in spatial data has statistical dependency to points around, however, the model has certain limitation on number of dependencies. In order to model complex spatial data, the limitation needs to be relaxed.

In the reliability literature, there are many models with covariates that conduct reliability analysis. The existing research can be divided into two distinct types: 1) parametric models and 2) semi-parametric models. In the parametric models the probability distribution of survival times and the overall shape of the hazard function need to be specified. Accelerated failure time [2], Weibull regression [20] and log-logistic regression [21] are parametric models in the literature. Recently, a more complex parametric failure

model considering the image of a material is proposed [22]. The disadvantage of the parametric models is that the prior knowledge about probability distribution may not be available.

In semi-parametric models, there are no assumptions on hazard functions, and they have parametric forms concerning the effect of the covariates. The advantage of this type of model is that it does not require prior knowledge about the form of true hazard functions (which can be very complex) to assess the effect of the covariates. The proportional hazard model (PHM) [23] is semi-parametric to analyze survival data [24]. The Cox model [24] is a well-known model based on PHM. Various models based on the Cox model have been proposed to select most significant covariates [25, 26]. Sleeper et al. [27] developed an approach based on the Cox model to capture the effect of covariates by smooth nonlinear B-splines. This model, however, may have many parameters and thus can suffer from overfitting. Faraggi et al. [28] developed a more complex form of the Cox model, but the model assumptions are difficult to be satisfied in reality. Thus, it can suffer from overfitting when there are a limited number of samples. Furthermore, an additive PHM [29] was proposed to model the effects of baseline hazard function additively rather than multiplicatively in the Cox model. Badia et.al [30] proposed a mixed model which considers additive and multiplicative effects simultaneously.

Although semi-parametric and parametric reliability models are being successfully applied on survival data to predict failure times, the models have strong assumptions regarding covariate relationships, which make the models ineffective in the case of complex covariates.

Furthermore, in literature, the parameters of reliability models are estimated using covariates and corresponding failure time (training process), and then predict failure time of new covariates (test process). However, if the distribution of training covariates and test covariates are not same, the performance of the model may be biased. This distribution discrepancy often neglected which may lead to inaccurate reliability estimation.

1.3 Dissertation Objective

In this dissertation research, we study on a deep learning-based reliability model for complex covariates. Especially, we first model the spatial covariates by a novel statistical and next we study on deep learning-based reliability model to predict future failure. Finally, a transfer learning-based reliability model is proposed.

The objectives of this research are listed as follows:

a) Develop an efficient spatial statistical approach to model complex covariates.
b) Apply the model in (a) to extract the feature of covariates and develop a deep learning-based model to predict future failure time
c) Develop a transfer learning model to predict reliability of materials/products using result of (a) and (b)

1.4 Dissertation Organization

This dissertation consists of three main chapters, preceded by the introduction chapter (i.e., CHAPTER 1) and followed by a general conclusion (i.e., CHAPTER 5). Specifically, in CHAPTER 2 a novel spatial statistical model is proposed to model complex and high dimensional covariates. In CHAPTER 3, a novel deep-learning based reliability model is

proposed to capture complex relationship of covariates and their failure time. Finally, in CHAPTER 4, a novel transfer learning-based reliability model is proposed for estimate reliability of products/materials using failure times of other products/materials.

CHAPTER 2. A Penalized Autologistic Regression

2.1 Overview

Recently dual phase high strength steel has attracted increasing attention in the automotive industry due to its prominent physical and mechanical properties. Microstructures of dual phase high strength steel have a significant effect on the properties of steel, such as wear resistance and strength, so it has an important role in the quality of steel. Therefore, statistical modeling of the microstructures of steel is of great interest. However, most existing methods require many model parameters due to the complex topological forms of microstructures, which make these models suffer from overfitting and high computational time for parameter estimation. To overcome these challenges, a novel statistical model is proposed to characterize microstructures and select the most effective parameters. Furthermore, an efficient parameter estimation method is developed to estimate the model parameters given a microstructure sample. The developed method is based on a penalized pseudo log-likelihood and the accelerated proximal gradient. A simulation study is conducted to verify the developed methods. The proposed methodology is validated by a real-world example of the microstructures of high strength steel, and the case study shows the superior performance of the developed model compared with existing methods.

2.2 Introduction

The main goal of the chapter to develop a statistical approach to model complex and high dimensional structure of covariate. The covariate is factor that effect the failure time. Existing methodologies to model complex spatial covariates have limitation on number of parameters to model the spatial data efficiently. To overcome the challenge, we proposed

a novel statistical model. The model can be used to reduce dimension of spatial data efficiently.

The motivation of this chapter is in the quality control of DP steel. Common quality control methods for steel manufacturing are traditionally based on appearance of the steels. However, research has shown that the microstructure of the steel plays an important role in mechanical properties. New statistical-based quality control methods [11, 19] require accurate microstructure quantifications. Our proposed model can be used for proper and accurate microstructure quantification within a class of steel or among different steel classes. Furthermore, another application of the proposed model in material science is enhancing the design and discovery of novel steels. Steel properties depend on grain size, location and orientation, which makes the number of possible configurations exponential and very costly to synthesize [31]. Moreover, human error is involved in the process. To overcome these challenges our proposed model can be utilized to design the desired steel properties efficiently by choosing the proper configuration of grains. Also, the proposed model can be used for microstructure reconstruction.

This chapter is organized as follows: After the introduction, Section 2.3 proposes the novel penalized autologistic regression model. Section 2.4 develops the parameter estimation method. Section 2.5 reports on simulation studies conducted to verify the proposed methodology, and Section 2.6 provides a real-world example of high strength dual-phase steel to illustrate the performance of the developed model. Finally, the chapter is concluded in Section 2.7.

2.3 Model description

In this study, the microstructure of two-phase materials is represented by a binary lattice denoted by $\mathbf{X}$, which is assumed to have a dimension of $d \times d$. Let $x_i \in \{0,1\}$ denote the observed value at the i^{th} site on a binary lattice, $i = 1,...,n$, where $n = d \times d$. Let $N(i)$ denote the collection of sites that are spatial neighbors of site i for a given neighborhood structure, and $x_j \in \{0,1\}, j \in N(i)$ represents the observed value of the j^{th} neighbor of site i. The cardinality of $N(i)$ is p, i.e., $|N(i)| = p$.

2.3.1 Introduction to the autologistic regression model

The autologistic regression model [32] has been widely used in the literature to study binary spatial data and is applied in multiple domains, including ecology, agriculture, epidemiology, and image analysis. The model assumes the probability that a site belongs to a phase only depending on its neighbors. This model property can be seen essentially as a Markov property in the Markov random field model.

Specifically, for the classical autologistic model, the conditional distribution of the site i depending on its neighbors $N(i)$ is defined as follows:

$$p(X_i = x_i \mid X_j = x_j : j \in N(i)) = \frac{\exp(\eta x_i)}{1 + \exp(\eta)} \tag{1}$$

$$\eta = \lambda_0 + \sum_{j \in N(i)} \lambda_j x_j$$

where $\lambda_j, j \in \{0,..., p\}$ is the model parameters.

To estimate the model parameters, the likelihood function of the classical autologistic regression model is given as follows [32] :

$$l(\boldsymbol{\lambda} \mid \mathbf{X}) = \frac{1}{c(\boldsymbol{\lambda})} \exp\left(\sum_i [\lambda_0 x_i + \frac{1}{2}(\sum_{j \in N(i)} \lambda_j x_i x_j)] \right) \quad (2)$$

where $\boldsymbol{\lambda} = \{\lambda_0, \lambda_1, \lambda_2, ..., \lambda_p\}$, $\mathbf{X}$ is the binary lattice, and $c(\boldsymbol{\lambda})$ is a normalizing function which is called the partition function.

2.3.2 A novel penalized autologistic regression model

The autologistic regression model incorporates spatial autocorrelation by considering the relationship between sites and their neighbors. The definition of order of neighboring is flexible in different contents [33]. As shown in Figure 2, the first order neighbors of site i are defined as $N(i) = \{8,12,13,17\}$, the second order neighbors are defined as $\{7,8,9,12,13,16,17,18\}$, the third order neighbors are defined as $\{3,7,8,9,11,12,13,14,16,17,18,22\}$, and the fourth order neighbors are defined as $\{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24\}$.

In the literature, the existing autologistic regression models generally consider neighboring up to the second order to avoid computational complexity [16]. High order of neighboring can significantly increase the number of model parameters, making the model suffer from overfitting and computationally difficult.

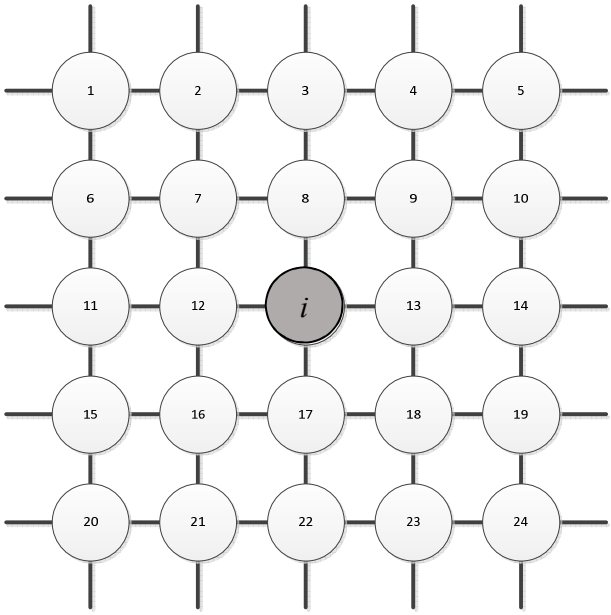

Figure 2. A microstructure image with 24 neighbors

To overcome the aforementioned challenges, we propose a novel penalized autologistic model to select the most relevant parameters so that high order of neighboring can be incorporated. The model is formulated as follows:

$$\max_{\lambda} \frac{1}{c(\boldsymbol{\lambda})} \exp\left(\sum_{i} [\lambda_0 x_i + \frac{1}{2} (\sum_{j \in N(i)} \lambda_j x_i x_j)] \right) \tag{3}$$

$$\text{Subject to } \lambda_0 + \sum_{j \in N(i)} |\lambda_j| \leq \alpha$$

where $\alpha \geq 0$ is a tuning parameter, and $\boldsymbol{\lambda} = \{\lambda_i\}$, $i = 0, 1, ..., p$ are model parameters.

The proposed model selects only relevant parameters so that it is effective for high dimension data. The constraint $\lambda_0 + \sum_{j \in N(i)} |\lambda_j| \leq \alpha$ in model (3) encourages the autologistic regression model so that some of the parameters become zero. The tuning parameter α controls the amount of shrinkage. The proposed model is able to simultaneously perform variable selection and parameter estimations.

In model (3), the introduced constraint makes the optimization problem difficult. It is convenient to consider the Lagrangian form of the optimization problem (3) [34] which is as follows:

$$\max_{\lambda} \frac{1}{c(\boldsymbol{\lambda})} \exp\left(\sum_{i} [\lambda_0 x_i + \frac{1}{2} (\sum_{j \in N(i)} \lambda_j x_i x_j)] \right) - \beta(\lambda_0 + \sum_{j \in N(i)} | \lambda_j |) \quad (4)$$

where β is a tuning parameter. Equations (3) and (4) are equivalent in a sense that for $\beta \geq 0$ there exists $\alpha \geq 0$, which results in the same solutions for the two equations.

The proposed penalized autologistic regression model (4) is a generalization of two classical models.

1) The proposed model generalizes the penalized logistic regression [35]. Specifically, our proposed model considers the neighbor of each site (i.e., auto-logistic model type), while the penalized logistic regression does not.

2) The proposed model is a generalization of the autologistic regression model. When $\beta = 0$ in (4), the proposed model degenerates to the likelihood function of the classical autologistic regression model.

To ensure that all sites have the same number of neighbors, we assume that the left top corner is connected with the right bottom corner and the right top corner is connected with the left bottom corner. In addition, the leftmost column is connected with the rightmost column, and the first row is connected with the last row.

2.4 Parameter estimation method

Given the model presented in the previous section, there are some challenges in model parameter estimation. In Section 2.4.1, we first develop the penalized pseudo-log

likelihood to overcome the challenge of dealing with the intractable computation of constant $c(\boldsymbol{\lambda})$ in the likelihood function (4). The next challenge is that traditional optimization methods are inefficient if not impossible to estimate model parameters. To overcome this challenge, we develop both an exact and an approximate algorithm based on the accelerated proximal gradient to estimate the model parameters. The developed exact and approximate methods are detailed in sections 2.4.2 and 2.4.3, respectively.

2.4.1 Penalized pseudo-log likelihood

The first challenge in model parameter estimation is dealing with the intractable computation of the constant $c(\boldsymbol{\lambda})$ in equation (4). Given a $d \times d$ microstructure image, we have to enumerate all $2^{d \times d}$ possible realizations of the image to calculate the normalization constant $c(\boldsymbol{\lambda})^{-1}$. To overcome this challenge, we adopt the pseudolikelihood approximation [14], in the autologistic regression framework and develop a penalized pseudo-log likelihood (PPLL) function as follows:

$$l_{PPLL} = \sum_{i=1}^{n} \log \frac{\exp(x_i(\lambda_0 + \sum_{j \in N(i)} \lambda_j x_j))}{1 + \exp(\lambda_0 + \sum_{j \in N(i)} \lambda_j x_j)} - \beta(\lambda_0 + \sum_{j \in N(i)} |\lambda_j|) \quad (5)$$

where β is a tuning parameter. Note that if $\beta = 0$, then (5) converts to a traditional pseudo-log likelihood function. To estimate the model parameters $\boldsymbol{\lambda}$, l_{PPLL} in (5) needs to be maximized.

2.4.2 Accelerated proximal gradient

As function (5) is not differentiable, the classical derivative-based optimization methods are not applicable. In addition, the classical non-derivative-based optimization methods

are not efficient to optimize equation (5). In this chapter, we develop a model parameter estimation method based on the accelerated proximal gradient framework [36].

The accelerated proximal gradient is a framework to solve optimization problems with a non- differentiable objective function. To utilize the framework, the objective function needs to satisfy two assumptions: 1) the objective function is a summation of a differentiable convex function $f(\boldsymbol{\lambda})$ and a non-differentiable convex function $g(\boldsymbol{\lambda})$, and 2) the differentiable part of objective function $f(\boldsymbol{\lambda})$ needs to be a Lipschitz continuous gradient.

Specifically, a function is a Lipschitz continuous gradient when there is a constant L that for every $\boldsymbol{\alpha}, \boldsymbol{\theta} \in \mathbf{R}^{p+1}$, the following inequality holds

$$\| \nabla f(\boldsymbol{\alpha}) - \nabla f(\boldsymbol{\theta}) \| \leq L \| \boldsymbol{\alpha} - \boldsymbol{\theta} \| \tag{6}$$

where $\|.\|$ is L2-norm, and $\nabla f(\boldsymbol{\lambda})$ is the gradient function.

2.4.2.1 Analytical solution

To apply the accelerated proximal gradient framework, the two aforementioned assumptions need to be satisfied. In this section we show that the optimization problem (5) satisfies both assumptions.

Following the maximum likelihood framework, to estimate model parameters, the l_{PPLL} function in (5) needs to be maximized. This is equivalent to minimizing $-l_{PPLL}$. Since the proximal gradient method focuses on minimization problems, we consider $-l_{PPLL}$ to be minimized. $-l_{PPLL}$ is a summation of the two separate functions, i.e., $f(\boldsymbol{\lambda}) + g(\boldsymbol{\lambda})$ which is formulated as follows:

$$f(\boldsymbol{\lambda}) = -\sum_{i=1}^{n} \log\left(\frac{\exp(x_i(\lambda_0 + \sum_{j\in N(i)} \lambda_j x_j))}{1+\exp(\lambda_0 + \sum_{j\in N(i)} \lambda_j x_j)}\right),\ g(\boldsymbol{\lambda}) = \beta \| \boldsymbol{\lambda} \|_1 \tag{7}$$

where $f(\boldsymbol{\lambda})$ is a convex differentiable function and $g(\boldsymbol{\lambda})$ is a convex and non-differentiable function. The following Proposition 1 shows that the second assumption is satisfied, and the proof of Proposition 1 is given in Appendix A.

Proposition 1: $f(\boldsymbol{\lambda})$ in equation (7) is a Lipschitz continuous gradient with $L = c\sqrt{(p+1)n}$, given a $d \times d$ microstructure sample image and $n = d^2$, and c is a constant larger than 1, i.e., $c > 1$.

As both assumptions are satisfied, we develop a proximal gradient method to optimize the objective function through an iterative algorithm. At each iteration, the model parameter is updated by

$$\boldsymbol{\lambda}^m = \arg\min_{\lambda}\left(f(\boldsymbol{\lambda}^{m-1}) + < \boldsymbol{\lambda} - \boldsymbol{\lambda}^{m-1}, \nabla(f(\boldsymbol{\lambda}^{m-1}) > + \frac{L}{2} \| \boldsymbol{\lambda} - \boldsymbol{\lambda}^{m-1} \|^2 + \beta \| \boldsymbol{\lambda} \|_1 \right) \tag{8}$$

where super-indices (m) denotes the iteration number, $<.>$ is the inner product operator, and $\|.\|_1$ is the L1-norm. However, the computational time of the parameter estimation method is high especially when the data dimension is high and high order of neighboring is considered. Furthermore, due to the complexity of the formula (8), the method needs additional time for a root-finding algorithm. In the next section we develop an approximated solution for (8) which takes less computational time.

2.4.2.2 Approximation solution

The analytical solution developed in the previous section can obtain the exact optimal solution. The computation time of the developed method, however, increases with the increment of the sample size. When the sample size is large, it may take too much time to obtain the analytical solution through the iterative method. We develop an approximation method to estimate the model parameters, which accelerates the optimization process.

The following Proposition 2 provides an approximated solution of the m^{th} iteration in (8), i.e., $\boldsymbol{\lambda}^m = \{\lambda_0{}^m, \lambda_1{}^m, ..., \lambda_p{}^m\}$ .The detailed proof of Proposition 2 is listed in Appendix B.

Proposition 2: The approximated closed form solution of each iteration m of (8) can be calculated as follows:

$$\lambda_j{}^m = \operatorname{sgn}\left(\frac{\sum_{i=1}^{n}(x_i x_j - x_j)}{2L} + \frac{(\lambda_j{}^{m-1})}{2}\right)\left(\left|\frac{\sum_{i=1}^{n}(x_i x_j - x_j)}{2L} + \frac{(\lambda_j{}^{m-1})}{2}\right| - \frac{\beta}{2L}\right)_+$$

where $\lambda_j{}^m$ is the j^{th} model parameters in the m^{th} iteration, $\operatorname{sgn}(.)$ is a sign function, $x_+ = \max(0, x)$ and $x_j = 1$ for $j = 0$.

The proposed optimization algorithm based on the accelerated proximal gradient algorithm and utilizing Proposition 2 to solve (8) is summarized in Algorithm 1 below:

Algorithm 1

1. Initialization of algorithm parameters; error size ε, iteration counter $m=1$, Lipschitz continuous gradient constant $L=c\sqrt{(p+1)}d^2$, model parameters initialization $\lambda_j^{(0)}$, auxiliary variable $\mathbf{y}^0=\{y_0{}^0 y_1{}^0,...,y_k{}^0\}=\mathbf{1}$
2. In each iteration, update

$$\lambda_j{}^m=\operatorname{sgn}\left(\frac{\sum_{i=1}^{n}(x_i x_j-x_j)}{2L}+\frac{(\lambda_j{}^{m-1})}{2}\right)\left(\left|\frac{\sum_{i=1}^{n}(x_i x_j-x_j)}{2L}+\frac{(\lambda_j{}^{m-1})}{2}\right|-\frac{\beta}{2L}\right)_+$$

3. Update $y_j{}^m=\lambda_j{}^{m-1}+m(\lambda_j{}^{m-1}-\lambda_j{}^{m-2})$, and $m=m+1$
4. If $|\lambda_j{}^{m-2}-\lambda_j{}^{m-1}|>\varepsilon$ then go to step 2

   Else stop.

The time complexity of the proposed algorithm is less than the analytical solution because fewer operations are needed to compute the solution and the root-finding algorithm is no longer needed. The convergence rate of Algorithm 1 is $O(\frac{1}{M^2})$, where M is the maximum iteration of the algorithm. The proof of the convergence rate of Algorithm 1 is listed in Appendix C. In Algorithm 1, the tuning parameter β needs to be estimated. In this chapter, we apply the K-fold cross validation method that is widely used in the literature [37]. Based on the K-fold cross validation, we developed Algorithm 2 below to obtain the optimum tuning parameter β.

Algorithm 2:

1. Initialize the input with the microstructure image data consisting of n pixels divided into the K subsample.

1. The estimated parameters are obtained as a function of the tuning parameter $\beta \in [0, L]$ using Algorithm 1 described above, while omitting the i^{th} fold, where $i = 1,...,K$.

3. The fitted model is used to predict the values of the omitted i^{th} subsample, and the prediction error is computed against each choice of the tuning parameter using the following formula

$$Error = -\frac{1}{n}\sum_{i=1}^{n} x_i \log p(X_i = 1 \mid X_j = x_j : j \in N(i)) + (1 - x_i) \log p(X_i = 0 \mid X_j = x_j : j \in N(i)) \quad (9)$$

4. The tuning parameter is chosen as the value of β which minimizes the error term in (9).

Equation (9) measures the average sum of the value of each pixel multiplied by the probability of having the value $(0/1)$ predicted by the proposed model. Specifically, if a pixel has a value of $X_i = 1$, then the right summand becomes 0, and the left summand remains in place. On the other hand, if a pixel has the value of $X_i = 0$, then the right summand with the term remains in place, but the left summand becomes 0.

When the sample size is not large, the leave-on-out method can be used. Leave-one-out is a type of *K*-fold cross validation when $K = n$, where n is the number of pixels in

the microstructure. In this study, for the i^{th} subsample that is obtained by omitting the i^{th} site, the model is fitted by using the subsample data and estimate error (9).

2.5 Simulation study

A simulation study is performed to illustrate the performance of the proposed parameter estimation method developed in Section 3. In the simulation study, we generate a sample microstructure according to the proposed penalized autologistic regression model. To simulate the microstructure sample, we first generate sample parameters $\boldsymbol{\lambda} = \{\lambda_i\},\ i = 0,...,p$ from the standard normal distribution and normalize them into the range of $[-1,0]$. In this study, the neighbor set $N(i)$ of the i^{th} pixel contains 24 neighbors, thus $\boldsymbol{\lambda} = [\lambda_0, \lambda_1, \lambda_2, ..., \lambda_{24}]$. We assume the microstructures are non-homogeneous so that all the parameters may have different values. To demonstrate the generalization of the proposed methods, an arbitrary $\boldsymbol{\lambda}$ is used in the simulation study. Additionally, some of the parameter values including λ_0 are set as zeros.

Next, given the parameters $\boldsymbol{\lambda}$, we develop an algorithm to generate a microstructure's image inspired by the procedure proposed by Cross and Jain [18]. The main idea behind the simulation algorithm is to increase the likelihood of the image's realization by repeatedly exchanging two randomly chosen pixel values. Specifically, we randomly choose two sites and reverse their values. The exchange is accepted if the new image gets a higher pseudo-likelihood value. The details of the simulation process are described in Algorithm 3, which is shown in Appendix D.

Based on Algorithm 3, we generate a binary matrix, with a sample size of 100×100, by using the given parameter $\boldsymbol{\lambda}$. Next, we apply Algorithm 1 to estimate the model

parameters, where the initial parameters are set as $\hat{\boldsymbol{\lambda}}_0 = [0, 0, ..., 0]$, and the error size is set as $\varepsilon = 0.001$.

When applying Algorithm 2, the tuning parameter β is obtained as $\beta = 50$. Based on Algorithm 1, the model parameters are obtained and compared with true values to access the performance of the proposed method.

Table 1 represents the estimated model parameter and the true values of the parameters. As shown in Table 1, the estimated parameters and the true values of the parameter are close, which demonstrates the performance of the proposed method. Additionally, the λ_0 is estimated as zero.

Table 1. True values and estimated model parameters

Parameters	True value	Estimation	Parameters	True values	Estimation
λ_1	-0.08	-0.091	λ_{13}	-0.09	-0.091
λ_2	-0.04	-0.056	λ_{14}	-0.2	-0.273
λ_3	-0.06	-0.075	λ_{15}	-0.001	-0.003
λ_4	-0.12	-0.184	λ_{16}	0	0
λ_5	-0.28	-0.34	λ_{17}	0	0
λ_6	-0.035	-0.047	λ_{18}	-0.06	-0.085
λ_7	0	-0.003	λ_{19}	-0.2	-0.258
λ_8	0	0	λ_{20}	0	0
λ_9	-0.1	-0.127	λ_{21}	0	0
λ_{10}	-0.2	-0.303	λ_{22}	0	0
λ_{11}	-0.01	-0.018	λ_{23}	-0.07	-0.090
λ_{12}	0	0	λ_{24}	-0.2	-0.258

The following Figure 3 shows the convergence of a randomly selected parameter, λ_6. As can be seen from Figure 3, the developed algorithm converges to the true value after around the 13^{th} iteration. Other parameters converge quickly as well.

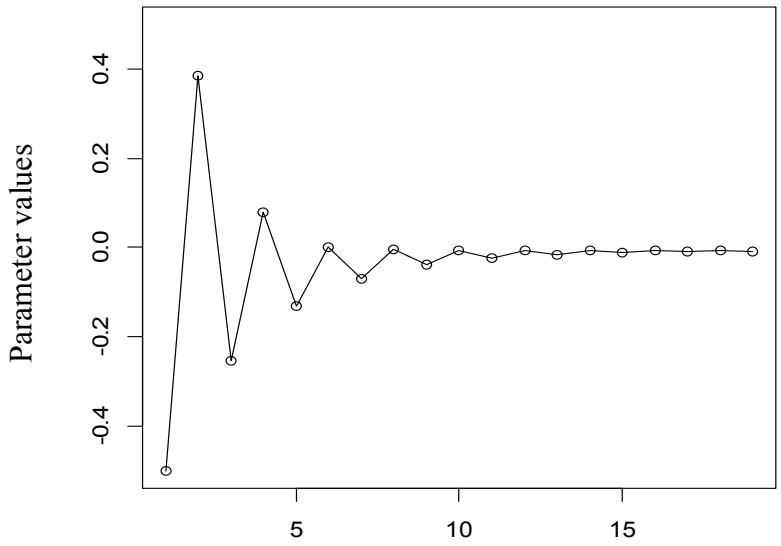

Figure 3. Parameter estimates vs. iteration number

Furthermore, we investigate the effect of sample size on the accuracy of the proposed method. To measure the performance of the developed method, we calculate the root mean square error (RMSE) [38], which is defined as follows:

$$RMSE = \sqrt{\frac{\sum_{i=0}^{p} (\hat{\lambda}_i - \lambda_i)^2}{p+1}} \tag{10}$$

where $\hat{\lambda}_i$ and λ_i are the i^{th} estimated model parameter and the true value of the parameter, respectively, and $p+1$ is the cardinality of $\boldsymbol{\lambda} = \{\lambda_0, \lambda_1, \lambda_2, ..., \lambda_p\}$.

The following Figure 4 shows the RMSE for different sample sizes. It can be seen in Figure 4 that the value of RMSE decreases as the size of the samples increases.

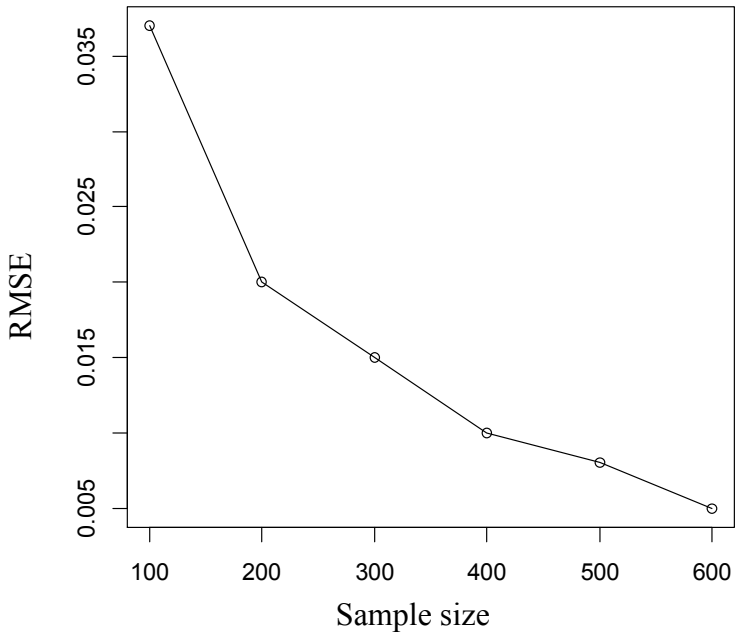

Figure 4. Sample size vs parameter estimations

Moreover, in order to analyze how many neighbors are falsely included or excluded during the parameter estimation methodology, we randomly generate 100 microstructure images with the dimensions 100×100 using Algorithm 3 and estimate model parameters based on Algorithm 1. Figure 5 shows the distribution of the false positive and false negative estimated parameters. Traditional autologistic regression estimates non-zero parameters in our simulation study, as no penalty term is involved in the model. Hence, the probability of a false negative for the 9 zero-value parameters is 1.

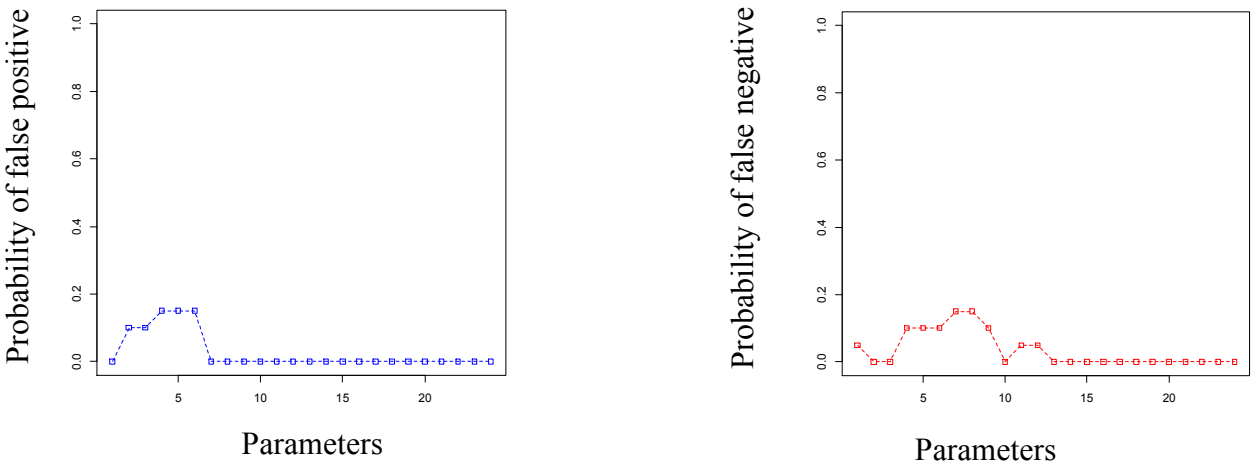

Figure 5. Distribution of false positive (left) and false negative (right)

Furthermore, we compare the computational time of the developed penalized autologistic regression model to the classical autologistic regression model in Figure 6. In the figure, the blue line shows the computation time of the classical autologistic regression model; the green and the red lines represent the computation time of the developed analytical method and the approximate method for the proposed penalized autologistic regression model, respectively. As shown in Figure 6, when the size of the sample is large, it is intractable to estimate the model parameters for the classical autologistic regression model, while our developed methods can significantly reduce the computational time of model parameter estimation.

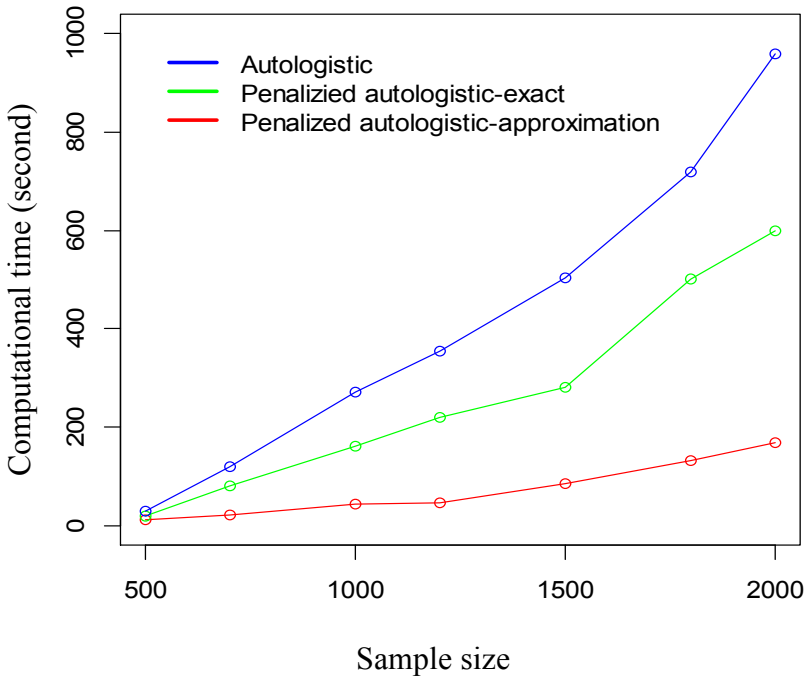

Figure 6. Sample size vs. computation time

To further analyze model accuracy, we compare estimated model parameters using penalized autologistic and traditional autologistic with second order of neighboring using their corresponding RMSE. Table 2 shows the RMSE of the penalized autologistic regression is less than that of the autologistic regression method. Thus, we can conclude that our proposed model outperforms the traditional model.

Table 2 Model fitting comparison

Model	**RMSE**
Penalized autologistic regression	0.10
Autologistic regression	0.37

2.6 Case study

We applied our proposed model to the microstructure images of samples of DP high strength steel, which is widely used in the automotive industry due to its excellent performance. The steel samples are prepared through several processing steps including grinding, etching and polishing. After preparation, the microstructures of the steel samples are obtained by using a microscope with a $1000X$ magnification.

We obtained 22 microstructure images of two types of DP steel, which are termed as DP780 and DP980, provided by our industrial partners. The image dataset includes 11 microstructure samples of DP780 steel and 11 microstructure samples of DP980.

Figure 7 shows the DP780 and a DP980 microstructures used in the case study. It can be seen in Figure 7 that the spatial distributions of the two phases are different in DP780 and DP980 microstructures.

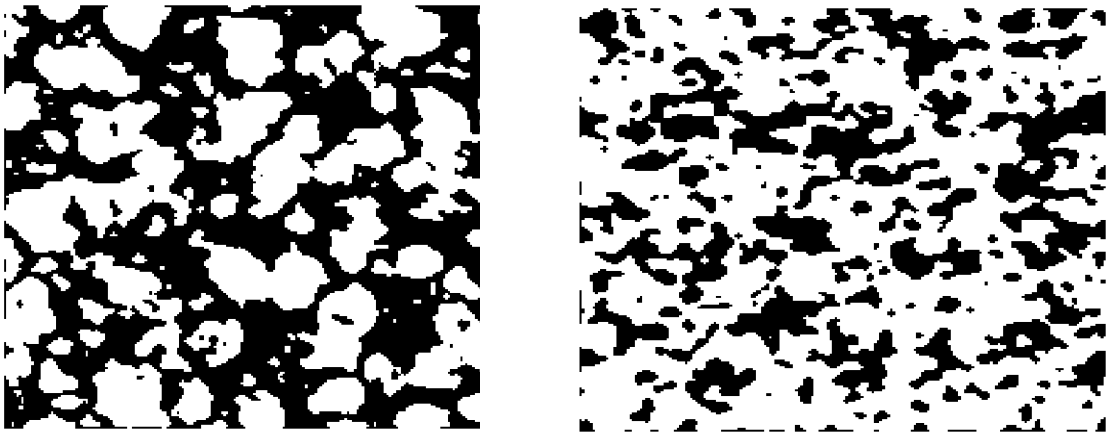

Figure 7. Two microstructure samples, a sample of DP780 (left) and a sample of DP980 (right)

One application of our proposed model is material classification from their microstructure images without any prior knowledge. However, it is a challenging task due to the large product variation that can appear across different manufacturers or different

batches from a manufacturer, resulting in variations on their microstructure. Irani and Taheri [39] showed that the product variations in the steel manufacturing process happen due to high variations of process parameters, e.g., heating energy and cooling speed. The product variations are physically presented on their microstructure. The sample variation, including both the variation of the mechanical and physical properties due to variation in the microstructures in the steel manufacturing process, is widely acknowledged and studied in the steel industry. Therefore, automatically classifying different types of steel products that exhibit variation is a challenging task.

In the case study, we investigate the problem of classifying two types of DP steel, i.e., DP780 and DP980, based on their microstructures. There is a large body of research in material science that characterizes and classifies material based on optical and electron imaging [40]. Despite the existing material variations, our proposed model is able to characterize materials based on their microstructure images considering their spatial properties in different phases.

The framework of classifying two types of microstructure samples consists of two steps. In the first step, features are extracted using a feature extraction method. In the second step, a classification method is used to classify microstructure images based on the extracted features.

In this chapter, parameters of the proposed penalized autologistic regression model can be used as features, based on which a classification technique is employed to classify two different types of DPs. The proposed model with the fourth order of neighboring is considered for the case study, and the model parameters are $\boldsymbol{\lambda} = [\lambda_0, \lambda_2, ..., \lambda_{24}]$. In this case, we assume that the steel is non-homogeneous so that the values of the model

parameters λ_i for different neighbors are different. We estimate each λ_j corresponding to the j^{th} neighbor for $j = 1,...,24$ and λ_0, based on Algorithm 1. The following Table 3 shows the model parameters of both a DP780 sample and a DP980 sample. In Table 3, some of the model parameters' values are zeros, which indicates that the parameters are unimportant in representing the underlying structure of the microstructure. Moreover, $\lambda_0 = -0.044$ for case of DP980 and $\lambda_0 = -0.078$ for case of DP780.

Table 3. Predicted parameters

Parameters	DP780	DP980	Parameters	DP780	DP980	Parameters	DP780	DP980
λ_1	-0.008	-0.010	λ_9	0	0	λ_{17}	0	0
λ_2	-0.004	-0.005	λ_{10}	-0.004	-0.007	λ_{18}	0	0
λ_3	-0.001	0.003	λ_{11}	-0.004	-0.005	λ_{19}	-0.005	-0.006
λ_4	-0.002	-0.006	λ_{12}	0	0	λ_{20}	-0.006	-0.010
λ_5	-0.006	-0.010	λ_{13}	0	0	λ_{21}	-0.003	-0.005
λ_6	-0.005	-0.007	λ_{14}	-0.003	-0.006	λ_{22}	-0.002	-0.003
λ_7	0	0	λ_{15}	-0.004	-0.007	λ_{23}	-0.004	-0.005
λ_8	0	0	λ_{16}	0	0	λ_{24}	-0.008	-0.010

After obtaining the estimated model parameters, the next step is to use an unsupervised classification method to classify the images. We use the *k*-means method [41], which is a classification method commonly used in the literature. The *k*-means method partitions m observations to *k* clusters in which each observation belongs to a cluster with the nearest mean. For our case, we set $m = 22$ and $k = 2$.

We applied the proposed model, the classical autologistic regression model, and the Gray Level Co-occurrence (GLC) method [42], to extract the features from the microstructure images and compared their performance by applying the *k*-means method in the second step. The GLC method that extracts image textural features including inertia, homogeneity, entropy and energy is the feature extraction method most commonly used in the literature [43].

Table 4 shows the classification accuracy (i.e., the percent of correct classification) of the *k*-means method with different feature extraction methods. As shown in Table 4, the classification method with features extracted by using the proposed penalize autologistic regression outperforms the classical autologistic regression model and the GLC method. The *k*-means method with our proposed model is able to classify all microstructure images correctly.

Table 4. Performance comparison of proposed model to the existing methods

Feature extraction method	**Classification accuracy of *k*-means method (%)**
Classical autologistic regression	10%
GLC	50%
Penalize autologistic regression	100%

2.7 Conclusion

In this chapter, we proposed a novel penalized autologistic regression model to characterize dual phase microstructures. We further developed a penalized maximum

pseudo log-likelihood method to estimate model parameters given microstructure samples. Furthermore, the traditional optimization methods cannot be directly applied to estimate model parameters due to their computational inefficiency especially when the numbers of neighbors are large. Hence, we developed efficient optimization methods to estimate the model parameters. We conducted a simulation study to verify the proposed parameter estimation method. As a real-world case study, we performed parameter estimation on a dataset of microstructures images. The dataset contained two different types of DP steel microstructures. We used the estimated parameters of the microstructure combined with a classification method to classify different types of high strength DP steel based on their microstructures. The classification results based on the proposed model outperformed the existing methods.

As a future research topic, study of a penalized multi-phase model which considers more than two phases would be interesting. The proposed regularized autologistic regression model will be extended to materials with more than two phases.

Chapter3. Deep Learning-Based Reliability Method for Complex Survival Data

3.1 Overview

Reliability of products is a critical issue as it has as high economic impacts, especially in current competitive markets. In modern applications, the complex and high dimensional data of products are collected which can be used for reliability analysis and the failure prediction. The existing reliability approaches, however, cannot efficiently model complex covariates and their effects on the time-to-failure of products. In this chapter, we propose a novel deep learning-based reliability approach to model the complex relationship of covariates and product failure. To estimate model parameters, neither the traditional deep learning parameter estimation method nor the maximum likelihood estimation method is applicable. To overcome this difficulty, a new model parameter estimation method is developed based on the partial likelihood framework. Furthermore, as there are often only a limited number of samples for real-world reliability problems, a new penalized partial likelihood estimation method is developed for this special circumstance. The developed method is capable of estimating model parameters for censored reliability data. A simulation study is conducted to verify the developed methods. The proposed method is justified by a real-world case study of the reliability analysis of materials. The case study shows that the proposed model outperforms the existing ones.

3.2 Introduction

Reliability estimation of products has crucial applications in various industries, particularly in current competitive markets, as it has high economic impacts. Hence, reliability analysis and failure prediction are receiving increasing attention. Reliability

models based on lifetime data have been developed for different modern applications. These models are able to predict failure by incorporating the influence of covariates on time-to-failure. The covariates are factors that affect the subjects' lifetime.

With the development of sensor technologies, high dimensional and more complex data can now be collected. These data can be used as covariates to predict the lifetime more precisely. For example, the advanced optical microscope can produce complex and high dimensional images for the material surface factors that affect materials' lifetime. Magnetic resonance imaging (MRI) and electronic health record (EHR) are highly complex covariates for patients' survival time. Existing reliability models, however, cannot efficiently model the effects of complex covariates on failure time. This chapter focuses on developing a novel reliability model to overcome this challenge.

To overcome existing models' limitations, we propose a semi-parametric deep learning-based reliability model in this chapter. The proposed model is an extension of the Cox model, and the same extension can be applied to other semi-parametric models. Unlike PHM, the proposed model does not assume a specific relationship between covariates and time-to-failure; rather, a deep artificial neural network (ANN) is trained to learn the complex and nonlinear relationship. Deep ANN includes representation of the learning algorithms that transform raw data to higher-level abstraction through a deep ANN containing a multi-level architecture. ANN is a parametrical model inspired by biology. The effectiveness of the model has been examined empirically and successfully applied in many fields including pattern recognition [44], classification [45] and regression [46]. In this chapter, multilayer perceptron (MLP), which is the most common class of ANN, is used. In MLP all connections between layers are in one direction and have real valued

weights. The output of each neuron is computed in a two-step process: first. a weighted sum of the input of each neuron is calculated; next, an activation function is applied to the value of the summation function to trigger the output of the neuron. The most common activation function is the sigmoid function [47],

In this chapter, we develop a reliability model based on that of MLP, whose advantages over traditional models are as follows: 1) the does not assume any specific distribution for the data, 2) the model is able to approximate any function with an arbitrary error, and 3) the model is nonlinear, which makes it suitable to model a complex relationship among covariates and failure time in real-world data. The model parameters, which are the weights of MLP, are estimated by minimizing a loss function.

In this chapter, the traditional MLP model parameter estimation methods cannot be directly applied because there is no access to the output value of MLP. To overcome this challenge, we develop a model parameter estimation method based on the partial likelihood to estimate the parameters of MLP. In addition, the model parameter estimation method may suffer from overfitting when there are only a few samples available to estimate the parameters of MLP, which is a common situation in real-world reliability problems. To overcome this difficulty, we develop a model parameter estimation method based on the penalized partial likelihood estimation method. The developed method can estimate the parameters of MLP with right censored survival data. The proposed model is verified and illustrated by simulation and a real-world case study.

The chapter is organized as follows: After the introduction, 3.3 proposes the novel deep learning-based reliability model. 3.4 develops the parameter estimation method. Section 3.5 implements simulation studies to verify the proposed method, and in Section 3.6 a

real-world case study is conducted to show the performance of the developed model. Finally, the chapter is summarized in Section 3.7.

3.3 Reliability model

We introduce the definition and the traditional Cox model in Section 3.3.1. The newly proposed reliability model based on MLP is presented in Section 3.3.2.

3.3.1 Introduction of the traditional Cox model

In reliability analysis, the hazard function $h(t)$ for failure time T is the probability that a subject fails during a small time interval given that the subject has not failed up to the beginning of the interval time. Let $\mathbf{x}=\{x_1,x_2,...,x_p\}$ denote the covariates which are associated to a time-to-failure $t \geq 0$; the Cox model is formulated as follows:

$$h(t \mid \mathbf{x}) = h_0(t)\exp(\mathbf{x}\boldsymbol{\alpha}^T) \tag{11}$$

where $h_0(t)$ is the baseline hazard (11) function - the hazard function when $\mathbf{x}=0$, $\boldsymbol{\alpha}=\{\alpha_1,..,\alpha_p\}$ is the unknown regression coefficients that need to be estimated (model parameters), and $\boldsymbol{\alpha}^T$ in (11) denotes vector transpose of $\boldsymbol{\alpha}$. The Cox model imposes a regression-type structure on the hazard function that is the product of two components. The first component $h_0(t)$ captures the effect of failure time, and the second component $\exp(\mathbf{x}\boldsymbol{\alpha}^T)$ expresses the effect of the covariates associated with failure time. The covariates which do not depend on time can be variables such as heat, pressure, contamination, etc.

3.3.2 A novel deep learning-based reliability model

Although the Cox model is widely applied for reliability analysis and failure prediction, the model assumes that the link function has an exponential form and the covariates $\mathbf{x} = \{x_1, x_2, ..., x_p\}$ have a linear combination in (11) which may not be sufficient to model the complex relationships of covariate on time-to-failure. Moreover, the model assumes an exponential link function which may not be satisfied in reality. In this study, we extend the traditional Cox model to a more complex function. The proposed model is formulated as follows:

$$h(t \mid \mathbf{x}) = h_0(t) g(\mathbf{x} \mid \boldsymbol{\theta}) \tag{12}$$

where $\boldsymbol{\theta}$ is the model parameter, $h_0(t)$ is a baseline hazard function (when $\mathbf{x} = \mathbf{0}$) which has a positive value depending on time t. $g(\mathbf{x} \mid \boldsymbol{\theta})$ is a function that determines the effect of the covariates on time-to-failure. Moreover, $g(\mathbf{x} \mid \boldsymbol{\theta})$ has two properties: 1) $g(0 \mid \boldsymbol{\theta}) = 1$; and 2) $g(\mathbf{x} \mid \boldsymbol{\theta})$ has non-negative values.

The proposed model (12) is a generalization of the traditional Cox model, as the function $\exp(\mathbf{x}\boldsymbol{\alpha}^T)$ is a special case of $g(\mathbf{x} \mid \boldsymbol{\theta})$. $g(\mathbf{x} \mid \boldsymbol{\theta})$ in the proposed model is a universal function that represents a complex nonlinear relationship. In the proposed model, function $g(\mathbf{x} \mid \boldsymbol{\theta})$ is represented by an MLP. MLP universal approximation theory [48] shows that an MLP with a sigmoid activation function can approximate the nonlinear and complex function of $g(\mathbf{x} \mid \boldsymbol{\theta})$ with certain error. Figure 8 illustrates a generic MLP with z hidden layers and one output layer to estimate the function of $g(\mathbf{x} \mid \boldsymbol{\theta})$.

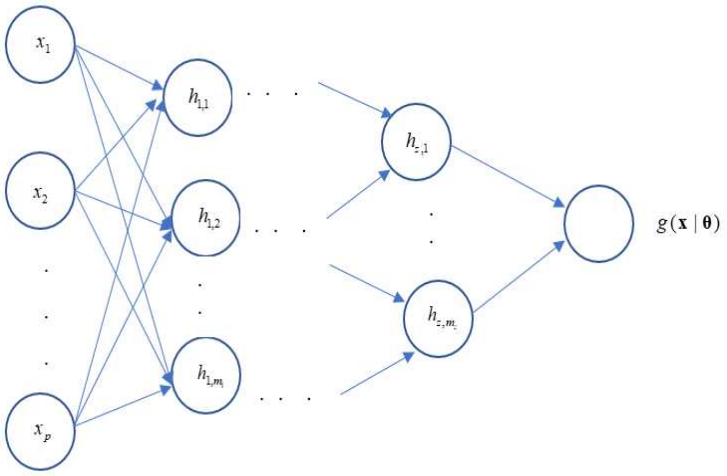

Figure 8. Structure of MLP with z hidden layers

In Figure 8, $\mathbf{x}=\{x_1,x_2,...,x_p\}$ is the input layer. The neural network has z hidden layer with $m_k, k=1,...z$ nodes in each hidden layer. $h_{i,j}$ is the j^{th} node in the i^{th} hidden layer and $g(\mathbf{x}\mid\boldsymbol{\theta})$ is the output node. Each $h_{i,j}$ is a neuron whose input is a weighted connection from the previous layer. The output of each neuron $h_{i,j}$ is formulated as follows:

$$\begin{aligned} h_{1,j} &= f\left(\sum_{i=0}^{p} x_i w^1{}_{i,j}\right); h_{k,j} = f\left(\sum_{i=0}^{m_{k-1}} h_{k-1,i} w^k{}_{i,j}\right); \\ g &= f\left(\sum_{i=0}^{m_z} h_{z,i} w^O{}_i\right), j=1,...,m_k,\ k=2,...,z \end{aligned} \tag{13}$$

where f represents the sigmoid activation function which is used as the activation function. $w^t{}_{i,j}, t=1,...,z$ is the parameter (weight) that connect the i^{th} neuron in the $(t-1)^{th}$ layer to the j^{th} neuron in the t^{th} layer, and $w^O{}_i$ denotes the parameters that connect the i^{th} neuron in the last hidden layer to the output layer. $h_{t,0}, t=1,...,z$ and x_0 denote the bias neuron and $w^t_{0,j}$ is bias parameters. The value of biased neurons is 1.

The bias allows for more variation which eventually causes richer representation of the input space to the learning model.

3.3.3 Properties of the proposed model

Based on the proposed deep learning-based reliability, the survival probability function $S(t \mid \mathbf{x})$ representing the probability of surviving at least *t* time units and the probability density function (PDF) can be derived as follows:

$$\begin{aligned} S(t \mid \mathbf{x}) &= \exp\left(-g(\mathbf{x} \mid \boldsymbol{\theta}) \int_0^t h_0(s) ds \right) = S_0(t)^{g(\mathbf{x}|\boldsymbol{\theta})} \\ f(t \mid \mathbf{x}, \boldsymbol{\theta}) &= h_0(t) S_0(t)^{g(\mathbf{x}|\boldsymbol{\theta})} \left(g(\mathbf{x} \mid \boldsymbol{\theta}) \right) \end{aligned} \tag{14}$$

where $S_0(t) = \exp\left(-\int_0^t h_0(s) ds \right)$ is the baseline survival function.

The proposed model is a semi-parametric model as all the instances share the same baseline hazard function, and the model parameter estimation is independent of the form of $h_0(t)$. Moreover, the model is a proportional hazard model since the hazard ratio (HR) is the same in all time points, where HR is defined as the hazard function of a sample's covariate $\mathbf{x}_i$ divided by a the hazard function of a different sample's covariate $\mathbf{x}_j$:

$$\frac{h(t \mid \mathbf{x}_i)}{h(t \mid \mathbf{x}_j)} = \frac{h_0(t)\left(g(\mathbf{x}_i \mid \boldsymbol{\theta}) \right)}{h_0(t)\left(g(\mathbf{x}_j \mid \boldsymbol{\theta}) \right)} = \frac{g(\mathbf{x}_i \mid \boldsymbol{\theta})}{g(\mathbf{x}_j \mid \boldsymbol{\theta})} \tag{15}$$

In the literature the Cox-Snell residual [49] is generally applied to assess the performance of the Cox model on data $(\mathbf{x}_j, t_j),\ j = 1,..,N$. The following Proposition 3 calculates the generalized Cox-Snell residual of the proposed model and assesses the performance of the proposed model. The proof of Proposition 3 is provided in Appendix E.

Proposition 3: The generalized Cox-Snell residual of the j^{th} sample is formulated as follows:

$$\hat{H}(t_j) = \hat{H}_0(t_j) g(\mathbf{x}_j \mid \boldsymbol{\theta}),\ j = 1, \ldots, N \tag{16}$$

where $\hat{H}_0(t_j)$ is the Breslow estimation of the baseline cumulative hazard function and is calculated as follows:

$$\hat{H}_0(t) = \sum_{t_i \le t} \frac{1}{\sum_{j \in R_i} g(\mathbf{x}_j \mid \boldsymbol{\theta})} \tag{17}$$

If the newly proposed deep learning-based reliability model fits the data well, then the generalized Cox-Snell residual follows an exponential distribution with $\mu = 1$.

Whether the generalized Cox-Snell residuals proposed in Proposition 3 follows exponential distribution can be verified by using the Kolmogorov-Smirnov test [50], which is a nonparametric test that compares a sample with a reference probability distribution.

3.4 Parameter estimation

Given the proposed model in the previous section, there are several challenges to estimate model parameters. First, since the baseline hazard function in (12) is not defined, the maximum likelihood estimation (MLE) cannot apply directly. Second, traditional MLP loss functions based on ordinary least square (OLS) are not applicable, as there is no access to the true values of $g(\mathbf{x} \mid \boldsymbol{\theta})$ during the training process. To overcome this challenge, a novel method based on the partial likelihood framework [51] is developed in section 3.3.1. Third, when there are few samples, which generally is the case with reliability problems, the parameter estimation method can suffer from

overfitting. To overcome this challenge, we develop the penalized partial likelihood in Section 3.4.2.

3.4.1 Loss function of MLP

To estimate the model parameters of the MLP, we use a loss function based on the partial likelihood function. We first formulate the conditional probability of the i^{th} sample's covariate $\mathbf{x}_i$ as follows:

$$p(\mathbf{x}_i \mid t_i) = \frac{g(\mathbf{x}_i \mid \boldsymbol{\theta}^T)}{\sum_{j \in R_i} g(\mathbf{x}_j \mid \boldsymbol{\theta}^T)} \tag{18}$$

where R_i consists of samples that their failure times are larger that t_i. This conditional probability only depends on the order in subjects that experience failure events. By assuming that the failure times of samples are independent, we can estimate the parameters of the MLP by maximizing the following log-partial likelihood function using (18):

$$\begin{aligned} \ell(\boldsymbol{\theta}) = \sum_{i=1}^{N} \left(\log\left(g(\mathbf{x}_i \mid \boldsymbol{\theta})\right) - \log \sum_{j \in R_i} \left[g(\mathbf{x}_j \mid \boldsymbol{\theta}) \right] \right) \\ s.t. \quad g(\mathbf{x}_i \mid \boldsymbol{\theta}) \geq 0 \;\; \forall \boldsymbol{\theta}, \mathbf{x}_i \\ g(0, \boldsymbol{\theta}) = 1 \quad \forall \boldsymbol{\theta} \end{aligned} \tag{19}$$

The constraints of (19) are due to the aforementioned two properties of $g(\mathbf{x}, \boldsymbol{\theta})$. The format of $g(\mathbf{x}, \boldsymbol{\theta})$ depends on the number of hidden layers of MLP, and the chosen activation function is formulated in (13).

The number of hidden layers and the number of neurons in each hidden layer depend on the input layer, the number of training samples and complexity of MLP network. General methods for determining the number of hidden neuron units are that it should be less than twice the size of the input layer [52].

3.4.2 Penalized partial likelihood estimation

In the case of small sample size, we develop a maximum penalized log-partial likelihood method to estimate the parameters of the MLP. Specifically, a penalty term is added to equation (19) so that MLP's loss function selects only the important weights to be estimated. The penalized log-partial likelihood loss function is formulated as follows:

$$\begin{aligned} \ell_p(\boldsymbol{\theta}) &= -\sum_{i=1}^{N}\left(\log[g(\mathbf{x}_i \mid \boldsymbol{\theta})] - \log\sum_{j\in R_i}\left[g(\mathbf{x}_j \mid \boldsymbol{\theta})\right]\right) \\ s.t. \quad & g(\mathbf{x}_i \mid \boldsymbol{\theta}) \geq 0 \quad \forall \boldsymbol{\theta}, \mathbf{x}_i \\ & g(0 \mid \boldsymbol{\theta}) = 1 \quad \forall \boldsymbol{\theta} \\ & \| \boldsymbol{\theta} \| \leq \delta \end{aligned} \tag{20}$$

where $\delta \geq 0$ is a tuning parameter, and $\boldsymbol{\theta} = \{\theta_i\}$, $i = 1, 2, ..., k$ are model parameters.

The developed method selects only the effective parameters of the developed model. The constraint $\| \boldsymbol{\theta} \| \leq \delta$ in (20) encourages some of the model parameters to become zero. The tuning parameter δ controls the amount of shrinkage (complexity of the model). The proposed model can simultaneously perform variable selection and parameter estimations. An equivalent and convenient form of (20) using the Lagrangian multiplier is formulated as follows:

$$\ell_p(\boldsymbol{\theta}) = -\sum_{i=1}^{N}\left(\log[g(\mathbf{x}_i \mid \boldsymbol{\theta})] - \log\sum_{j\in R_i}\left[g(\mathbf{x}_j \mid \boldsymbol{\theta})\right]\right) + \beta \| \boldsymbol{\theta} \|$$
$$s.t. \quad g(\mathbf{x}_i \mid \boldsymbol{\theta}) \geq 0 \;\; \forall \boldsymbol{\theta}, \mathbf{x}_i \qquad (21)$$
$$g(0 \mid \boldsymbol{\theta}) = 1 \quad \forall \boldsymbol{\theta}$$

where β is a tuning parameter and $\|.\|$ is norm 1. Equations (20) and (21) are equivalent in a sense that for $\beta \geq 0$ there exists $\delta \geq 0$, which results in the same solutions for the two equations. In this study, the tuning parameter β in equation (21) is estimated by the K -fold cross validation method [53] that is widely used in the literature.

Traditional derivative-based optimization methods are not applicable to solve (21) due to the non-differentiable part of the objective function $\beta \| \boldsymbol{\theta} \|$. In this study, a heuristic method, particle swarm optimization (PSO) [54] is chosen to solve (21). The PSO is a computational approach that optimizes problems by iterative attempts to improve candidate solutions for a given quality metric. It solves the problem by proposing a set of candidate solutions (here referred to as particles) based on simple mathematical formulas on the position and the velocity of the particles and moving them in the search space. The motion of each particle is affected by its locally best-known position and is directed to the most famous locations in the search space, which are updated as other particles find a better position. This is expected to push the swarm to the best solution.

The developed method can be extended to include censored data. Let D denote the indexes of subjects having failure times, and let S denote the indexes of the subject that are right censored at time t^* with $D \cup S = \{1, 2, ..., N\}$. The penalized log-partial likelihood method including right censored survival time is formulated as follows:

$$\ell_p(\boldsymbol{\theta}) = -\sum_{i=1}^{N} \delta_i \left(\log[g(\mathbf{x}_i \mid \boldsymbol{\theta})] - \log \sum_{j \in R_i} \left[g(\mathbf{x}_j \mid \boldsymbol{\theta}) \right] \right) + \beta \| \boldsymbol{\theta} \|$$
$$s.t. \quad g(\mathbf{x}_i \mid \boldsymbol{\theta}) \geq 0 \quad \forall \boldsymbol{\theta}, \mathbf{x}_i \qquad (22)$$
$$g(0 \mid \boldsymbol{\theta}) = 1 \quad \forall \boldsymbol{\theta}$$

where δ_i if the i^{th} subject's survival time is right censored and otherwise $\delta_i = 1$. The loss function introduced in (22) allows to estimate parameters of MLP in present of right censored survival data.

3.5 Simulation Study

In the simulation study we first generate a set of samples and failure times. Specifically, we generate samples consisting of covariates $\mathbf{x}_i$ which are drawn randomly from a standard normal distribution, and the covariates are chosen to have 4 elements, i.e. $\mathbf{x}_i \in R^p, p = 4$. Furthermore, $g(\mathbf{x} \mid \boldsymbol{\theta})$ is chosen to be a complex function which satisfies the aforementioned properties in Section 3.2, i.e., $g(\mathbf{0} \mid \boldsymbol{\theta}) = 1$ and $g(\mathbf{x} \mid \boldsymbol{\theta}) \geq 0$. The function is formulated as follows:

$$g(\mathbf{x} \mid \boldsymbol{\theta}) = \begin{cases} 1 & \mathbf{x} = 0 \\ \left| \sum_{i=1}^{2} \sin(x_i) + \sum_{j=3}^{4} \cos(x_j) \right| & \mathbf{x} \neq 0 \end{cases} \qquad (23)$$

where $|.|$ denotes the absolute value and x_j represents the j^{th} component of a sample. In this simulation study, we consider the Weibull lifetime distribution, which is commonly used in the reliability field [55]. Based on the Weibull lifetime distribution, the simulated failure time can be computed using inverse cumulative density probability function as follows:

$$T = \left(-\frac{\log(U)}{\lambda g(\mathbf{x}|\boldsymbol{\theta})} \right)^{1/\alpha} \tag{24}$$

where $\lambda, \alpha > 0$ are the scalar and shape parameters of Weibull distribution, respectively. Hence, the corresponding hazard function can be formulated as follows:

$$h(t|\mathbf{x}) = \frac{d}{dt}\left(g(\mathbf{x}|\boldsymbol{\theta}) H_0(t) \right) = \lambda \alpha t^{\alpha-1} g(\mathbf{x}|\boldsymbol{\theta}) \tag{25}$$

In this study scalar and shape parameters are chosen to be $\lambda = \alpha = 0.5$. We generate $g(\mathbf{x}_i|\boldsymbol{\theta})$ by random 100 covariates using equation (23), their corresponding time-to-failures using equation (24), and the corresponding hazard function using equation (25)

We consider two scenarios where sample sizes are large and small to illustrate the performance of partial likelihood and the penalized likelihood estimation method. For the first scenario (large sample size), we randomly select 50 samples to train with the corresponding time-to-failures. We apply the proposed MLP with 2 hidden layers of 2 neurons in the first and second layers. Then we minimize equations (19) and (21) to estimate the model parameters, i.e., the training process. Next, we evaluate the performance of the trained model on the remaining 50 covariates with the corresponding time-to-failures, i.e., the testing process.

To analyze the accuracy of the model, we compute the mean absolute error (MAE) [54] for the models of the 50 covariate test set. A Smaller ME indicates better performance. The formulation of MAE is given as follows:

$$MAE = \frac{1}{N} \sum_i | g(\mathbf{x}_i|\boldsymbol{\theta}) - \hat{g}(\mathbf{x}_i|\boldsymbol{\theta}) | \tag{26}$$

where N is the test set size; $\hat{g}(\mathbf{x}_i \mid \boldsymbol{\theta})$ is the estimated function of $g(\mathbf{x}_i \mid \boldsymbol{\theta})$ for the i^{th} sample and $|.|$ represents absolute value.

Table 5 shows that the MAE values of our proposed model with the partial likelihood parameter estimation method (named Deep Learning-Partial) and the penalized likelihood parameter estimation method (named Deep Learning-Penalized) are smaller than that of the traditional Cox model, indicating that our proposed model outperforms the Cox model. In addition, the table shows that the Deep Learning-Partial has the best performance for failure data with large sample size.

Table 5. Performance of proposed models and Cox model on large sample size

Method	**MAE**
Deep Learning-Penalized	0.38
Deep learning-Partial	0.21
Cox model	0.52

Furthermore, we computed the MAE for different training set sizes and measured the performance of the trained Deep Learning-Partial model on the test set with 50 samples. As shown in Figure 9 as training set size increases, the value of MAE decreases.

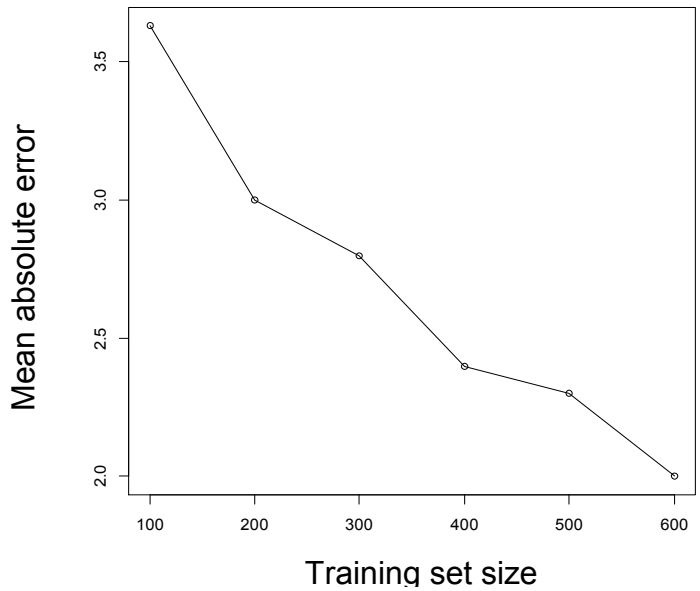

Figure 9. MAE vs training set size

Figure 10 shows the true hazard function, the estimated hazard functions of the Cox model, and that of our proposed deep learning-based reliability model whose model parameters are estimated by the partial likelihood method given a randomly selected covariate. As the figure shows, the estimated hazard function of our proposed model is very close to the true hazard function, and it outperforms the Cox model.

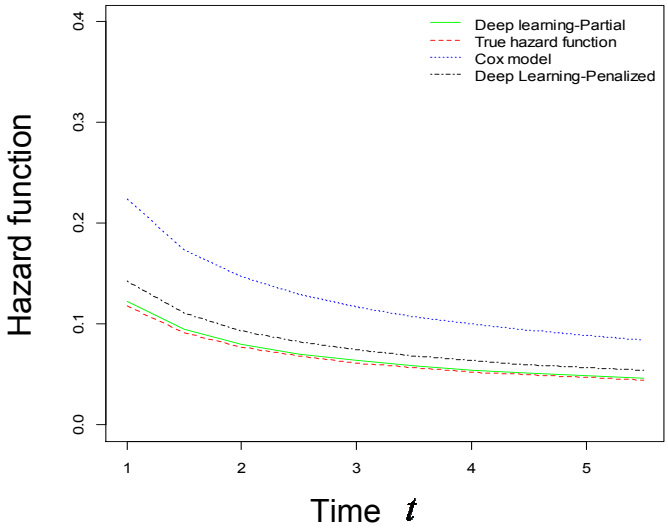

Figure 10. Hazard function vs time

For the second senario, we consider a small sample size. Specifically, we randomly select 25 samples to train and test the model by randomly selecting 25 samples. We apply the proposed model with Deep Learning-Partial, Deep Learning-Penalized and the traditional Cox model. The same MLP structure is used as explained in the first scenario. Table 6 shows the performance of the three models. As the table shows, the Deep Learning-Penalized has the best performance when a small sample size is available.

Table 6. Performance of proposed models and Cox model on small sample size

Method	**MAE**
Deep Learning-Penalized	0.27
Deep Learning-Partial	0.32
Cox model	0.48

To measure the effect of model complexity on prediction accuracy, we evaluated the performance of the proposed model for different values of β. The parameter β controls the model complexity. A small β makes the model more complex, and a larger β makes the model less complex. It has been shown in the literature that when a complex model is learned the training data's noise and suffers from overfitting. Similarly, a very simple model is not capable of modeling complex data, and the model suffers from underfitting [56]. A model that suffers from overfitting or underfitting has poor performance on a test set. Hence, choosing the right complexity is important in model performance.

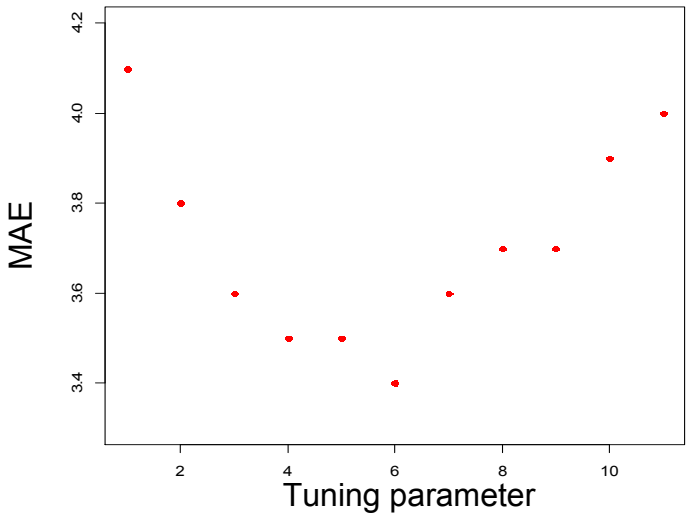

Figure 11. Effect of tuning parameter on the accuracy of the model

Figure 11 shows the performance of the Deep Learning-Partial model when a large sample size is available to estimate model parameters with different values of β (model

complexity). In case of small sample size, the Deep Learning-Penalized Partial model with different values of β has a similar trend.

3.6 Case Study

We apply our proposed model to DP steels to capture the complex effect of their microstructure.

We obtained 20 microstructure images of DP steel, which is called DP780, with the corresponding failure time, $t_i, i = 1,..,20$. A tensile test was conducted using the Instron 8801 testing machine to obtain the steel samples' failure times. The microstructure images are obtained using a microscope with a $1000X$ magnification after the steel sample was prepared with several preprocessing steps including grinding, chemical etching, and polishing. Moreover, the size of each microstructure image was 100×100. To apply our proposed model to the dataset, first we reduced the dimensions of the images by applying autologistic regression to extract the covariates [57].

Autologistic regression was developed to study the spatial binary data, and the model has been applied in multiple disciplines including ecology [15], agriculture [16] and image analysis [17]. An autologistic regression model assumes that the probability that a site belongs to 0 or 1 depends only on its neighbors, where neighbors are defined as a collection of image pixels around this site/pixel. Cross et.al [18] showed that the autologistic regression model is well suited for the binary image, which is relevant to the material microstructure modeling problem in this chapter.

In this study, we applied autologistic regression with two neighboring orders to characterize microstructures. The corresponding autologistic regression model considers

8 connecting neighbors $\{a_u, a_d, a_l, a_r, a_{ur}, a_{ul}, a_{dr}, a_{dl}\}$ [19]. Furthermore, we assume the DP AHSS materials are homogeneous and the microstructure images are anisotropic. Based on the assumptions $\{a_u, a_d\}, \{a_l, a_r\}, \{a_{ur}, a_{ul}\}$ and $\{a_{dr}, a_{dl}\}$ share their coefficients [19]. Hence, 5 covariates (coefficients) of each image were obtained including an intercept, i.e., $\mathbf{x}_i \in R^5$, $i = 1, ..., 20$. Table 7 shows the value of each covariate in a randomly selected microstructure image, i.e. x_1, x_2, x_3 and x_4. The value of intercept parameter is x_0=-0.0799

Table 7: Extracted covariates of a microstructure using autologistic regression

Covariates	Values	Covariates	Values
x_1	-0.0035	x_3	-0.0051
x_2	-0.008	x_4	-0.0065

Next, we applied our proposed MLP with 2 hidden layers of 2 neurons and 1 neuron of output. We trained the MLP using the penalized partial likelihood estimation method to predict the future hazard function. Also, we analyzed the proposed model to verify that the data fits the model using Proposition 1. Specifically, the Kolmogorov-Smirnov test verifies that the generalized Cox-Snell residuals follow exponential distribution. Specifically, the p-value of the test is 0.99, which shows that samples are drawn from exponential distribution. Therefore, the proposed model fits the data. The estimated model parameters (weights of MLP) are shown in Table 8 (the values include 10^{-10})

Table 8. Estimated model parameters

Parameters	Values	Parameters	Values
w_{11}^{1}	0.1	w_{42}^{1}	0
w_{21}^{1}	0.2	w_{52}^{1}	0
w_{31}^{1}	0.1	w_{11}^{2}	-0.1
w_{41}^{1}	0	w_{21}^{2}	0
w_{51}^{1}	-0.1	w_{12}^{2}	-0.3
w_{12}^{1}	-0.1	w_{22}^{2}	-0.1
w_{22}^{1}	0	w_{11}^{O}	0
w_{32}^{1}	0	w_{21}^{O}	0

As the table shows some of the estimated weights are zero, which makes the MLP sparse and prevents the model from overfitting when number of training set is not very large.

Furthermore, to analysis the computational time and convergence of the developed method, Figure 12 represents the convergence rate of three randomly selected MLP's weights $(w_{21}^{1}, w_{51}^{1}, w_{22}^{2})$ (shown in different colors i.e. blue, red and black) which are estimated by the developed method. As it can be seen from the Figure 6, the parameters

converge in the 250^{th} iteration. The other parameters converge at a similar number of iterations.

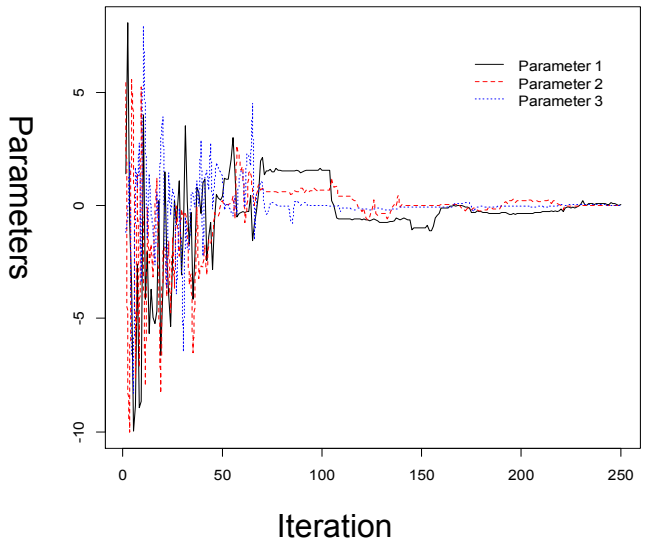

Figure 12. Convergence rate of parameters

To further analyze and demonstrate the advantage of our proposed model in terms of failure prediction, we compared the results of our model with existing models. Specifically, we used the Akaike's information criterion (AIC) to measure model fitting. AIC is formulated as follows:

$$AIC = -2(\log lik) + 2 * p \tag{27}$$

where p is the number of parameters.

A model fits the data best by minimum AIC [58]. As it is shown in Table 9, our proposed model has superior performance over the traditional Cox model.

Table 9. AIC measurement of Cox model vs deep learning-based reliability model

Model	AIC
Cox model	133.5
Deep Learning-Penalized	100.7

3.7 Summary

Modern sensors technologies produce complex covariates coupled with failure times. In this chapter, we proposed a novel deep leaning-based survival model to capture the effects of complex covariates on a material's failure. The proposed model is a generalization of the Cox model that traditionally assumes that linear combination of covariates and an exponential link function. In the proposed deep leaning-based reliability model, the assumptions are relaxed to be more flexible using MLP. To estimate model parameters, neither traditional MLP training methodology nor maximum likelihood estimation is applicable. To overcome this challenge, we developed a partial likelihood-based method to estimate the parameters of MLP. Also, we developed a penalized partial likelihood-based method to overcome the overfitting problem when the number of samples is small, which generally appears in reliability problems. Furthermore, a model parameter estimation was developed to train MLP with right censored survival data. A simulation study and a case study were implemented to verify the proposed methodology and superior performance compared with the tradition model.

As a future research topic, a study to generalizes the Cox model with time dependent covariates and directly apply ANN to estimate hazard function would be interesting.

Chapter 4. Transfer Learning-based Reliability Model with Complex Survival Data

4.1 Overview

Estimating the reliability of products has high priority in the current competitive market. Existing reliability models require failure times of the products to estimate model parameters and predict future failure times. However, obtaining the failure time of new products can be costly and time consuming in real-world applications, especially with high quality and reliable products. To overcome this challenge, we propose a semi-parametric transfer learning-based reliability model to utilize the covariates and failure time of similar products whose failure times are accessible. There are several challenges to estimate model parameters. First, the covariates have complex effects on failure time; second, the distribution of the covariates of new products are different from that of similar products or materials. To overcome these difficulties, we develop a parameter estimation method based on deep learning. Specifically, the developed method is based on a two-level autoencoder to transfer the covariates to a new distribution space by minimizing the distribution distance between the hidden layers of the autoencoders. Furthermore, a deep learning network is developed to capture the complex effect of the transferred covariates on failure times. A simulation study is conducted to verify the developed method. The proposed method is justified by a real-world case study of the reliability analysis of materials. The case study shows that the proposed model outperforms the existing ones.

4.2 Introduction

The reliability estimation of products is currently gaining increasing attention since it has a high impact in various applications. Reliability models can quantify product quality in mechanical applications or efficiency of a new treatment in healthcare applications.

Existing reliability models rely on failure time information to predict future failure time. However, obtaining failure time information is costly and time consuming. In mechanical applications, for example, to generate failure time, materials or products have to run costly tests for a long period of time. The issue is especially attracting increasing attention with high quality products available today. This chapter focuses on developing a novel reliability model to overcome this challenge by utilizing the failure time of similar subjects of interest.

Domain adaptation is used in problems when there is data from two related domains but under different distributions. Domain differences are the main obstacles to adoptable cross-domain predictive model. Our goal is to utilize the failure information from some existing material (source) and predict the failure time of a new set of materials (target).

Covariate-based reliability models traditionally estimate model parameters by source data, and then the target data is directly applied to predict failure time. The methods work well when the source and target data are from the same domain or follow the same distribution. However, in reality, source and target data are from different domains or distribution, and a domain shift (domain adoption) is needed.

In the area of domain adoption, there are two different types: 1) unsupervised domain adoption and 2) semi-supervised domain adoption. In unsupervised domain adoption, there is no available respond variable, and in semi-supervised domain adoption there are few respond variables available. John Blitzer [59] proposed a structural correspondence learning method that uses a pivot feature from the source and target to find the correspondence among the features. Daumé [60] proposed a heuristic kernel method to

adopt a target domain. A dimension reduction based model was introduced [61] to reduce the divergence between the source and target domain.

Although these reliability models have been successfully applied to predict failure time, there is no research on predictive failure time when there is no failure time information available for an object of interest. We propose a transfer learning-based reliability model that is an extension of the Cox model, and the same extension can be applied to other reliability models. Specifically, the proposed model is based on deep learning to transfer the source and target domains to a new destitution space such that the transferred covariates have the same distribution. A novel loss function is developed to estimate the model parameters.

Deep learning is a powerful method used in many applications. Deep learning uses a hierarchical architecture with non-linear units to capture the high-level information in observations. Deep learning is a suitable method for domain adoption and transfer learning [62, 63]. Fine-tuning deep neural network (DNN) architectures is popular in semi-supervised domain adoption [64]. To adopt the domain, Oquab et.al [65] proposed to train a DNN on a source domain and freeze part of the DNN's weights and add some layers to adopt it to a new (target) domain. Chu et al [66] explored the performance of fine-tuning DNN architectures across multiple target sets. Their main assumption is that the internal layers of the DNN can act as a generic extractor of mid-level image representation, which can be pre-trained on a source, but in reality, this assumption may not be true. Furthermore, Chen and Chien [67] proposed deep semi-supervised learning for domain adoption by introducing a multi task objective function, and Glorot et.al [68] proposed an unsupervised domain adoption method for sentiment classification.

In this chapter, we use autoencoder [69] to minimize distribution discrepancy by transferring the distributions of source and target domains to a new distribution space. An autoencoder is a type of deep learning which is used to learn efficient data coding in an unsupervised manner. An autoencoder learns to encode data from the input layer into a short code and then decode that code into something that closely matches the original data.

This chapter is organized as follows: After the introduction, Section 4.3 proposes our novel transfer learning-based reliability model. Section 4.4 develops the parameter estimation method. Section 4.5 reports on simulation studies conducted to verify the proposed methodology, and Section 4.6 provides a real-world example of high strength dual-phase steel to illustrate the performance of the developed model. Finally, the chapter is concluded in Section 4.7.

4.3 Methodology

We proposed a new adopted DNN for unsupervised domain adoption to predict failure time. The proposed model maps between source tasks to the target task. Let (x^s, y^s) denote the source domain information; $x^s \in R^p$ is the covariate of the source domain, and y^s is its corresponding failure time. Furthermore, (x^t, y^t) denote the target domain information; $x^t \in R^p$ is the covariate of target sample with unknown failure time. The proposed model is an extension of the traditional Cox model to include domain adoption. The proposed model is formulated as follows:

$$h(y^t \mid x^t) = h_0(y^s) g(x^s, x^t, \boldsymbol{\theta}) \tag{28}$$

where $\boldsymbol{\theta}$ is the model parameter, and $h_0(y^s)$ is a baseline hazard function when $\mathbf{x}=\mathbf{0}$. The baseline function is a positive value depending on time y^s. $g(x^s, x^t, \boldsymbol{\theta})$ is a function of the source covariates that are adopted by the target samples x^t. The function determines the effect of the covarriates on failure time. Moreover, $g(x^s, x^t, \boldsymbol{\theta})$ in (28) have two properties: 1) $g(x^s=0, x^t=0, \boldsymbol{\theta})=1$ and 2) $g(x^s, x^t, \boldsymbol{\theta})$ has non-negative values.

The proposed model is a semi-parametric model as all the instances share the same baseline hazard function, and the model parameter estimation is independent of the form of $h_0(t)$. Moreover, the model is a proportional hazard model since the hazard ratio (HR) is the same in all time points, where HR is defined as the hazard function of a sample's covariate $\mathbf{x}_i^t$ divided by a the hazard function of a different sample's covariate $\mathbf{x}_j^t$:

$$\frac{h(y_i^t \mid x_i^t)}{h(y_j^t \mid x_j^t)} = \frac{h_0(y_i^s) g(x_i^s, x_i^t, \boldsymbol{\theta})}{h_0(y_j^s) g(x_j^s, x_j^t, \boldsymbol{\theta})} = \frac{g(x_i^s, x_i^t, \boldsymbol{\theta})}{g(x_j^s, x_j^t, \boldsymbol{\theta})} \tag{29}$$

4.4 Parameter estimation

Given the proposed model in the previous section, there are several challenges to estimate model parameters. 1) $g(x^s, x^t, \boldsymbol{\theta})$, which reflects the effect of covariates on failure times, has a complex form; and 2) distributions of the target and source are different, which can affect the performance of prediction. To overcome these challenges, we developed a deep learning domain adoption framework based on an autoencoder and MLP to minimize the distribution distance of the source and target domains and estimate model parameters. Specifically, the framework consists of two level autoencoders whose inputs are source and domain covariates, and the outputs are transferred covariates with

minimum distribution distance of the domain and target domains. Figure 13 illustrates the detailed structure of the proposed framework.

To estimate the parameter of the proposed framework, we developed the new loss function, formulated as follows:

$$\sum_{i=1}^{N} \| \mathbf{x}_i^{\,s} - \hat{\mathbf{x}}_i^{\,s} \|_2^{\,2} + \sum_{i=1}^{N} \| \mathbf{x}_i^{\,t} - \hat{\mathbf{x}}_i^{\,t} \|_2^{\,2} + \sum_{i=1}^{N} \left(\sum_{k=1}^{z} MMD(h^t_{\,i,k}, h^s_{\,i,k}) + MMD(\hat{\mathbf{x}}_i^{\,s}, \hat{\mathbf{x}}_i^{\,t}) \right) \tag{30}$$

where $\mathbf{x}_i^{\,s}$ and $\mathbf{x}_i^{\,t}$ are the i^{th} sample of the source and target domain, respectively. $\hat{\mathbf{x}}_i^{\,s}$, $\hat{\mathbf{x}}_i^{\,t}$ are denoted as the corresponding predictions. $h^t_{\,i,k}$ and $h^s_{\,i,k}$ denote the k^{th} hidden layer of the autoencoders of the source and target domain, respectively. Furthermore, MMD represents the maximum mean discrepancy. MMD is a statistical test to determine if two samples are drawn from different distributions. The test statistic is the largest difference in expectations over functions in the unit ball of a reproducing kernel Hilbert space (RKHS). MMD is empirically computed as follows:

$$MMD^2(s,t) = \frac{2}{N} \sum_{i=1}^{N/2} h(z_i) \tag{31}$$

where h is an operator defined on a quad-tuple as follows:

$$\begin{aligned} h(z_i) &= k(x^s_{\,2i-1}, x^s_{\,2i}) + k(x^t_{\,2j-1}, x^t_{\,2j}) \\ &- k(x^s_{\,2i-1}, x^t_{\,2j}) - k(x^t_{\,2i}, x^t_{\,2j-1}) \end{aligned} \tag{32}$$

where $z_i = \{x^s_{\,2i-1}, x^s_{\,2i}, x^t_{\,2j-1}, x^t_{\,2j}\}$ and $s = \{x^s_{\,1}, ..., x^s_{\,n}\}, t = \{x^t_{\,1}, ..., x^t_{\,n}\}$. Furthermore $k(.,.)$ is a kernel function. MMD has a range between 0 and 1. When the distribution of two samples s and t are identical, the statistic is equal to zero; MMD=0.

$\hat{\mathbf{x}}_i^{\,s}$ and $\hat{\mathbf{x}}_i^{\,t}$ in (30) are transferred covariates which have close distribution. Next, the transferred source covariates and their corresponding failure times are applied to the proposed deep learning-based reliability model introduced in Chapter 3 to capture the complex effect of the covariates on failure times. The loss function to estimate the parameters of the MLP is based on the partial likelihood function. The loss function is formulated as follows:

$$\ell_p(\boldsymbol{\theta}) = -\sum_{i=1}^{N}\left(\log\left(g(\hat{x}_i^{\,s},\boldsymbol{\theta})\right) - \log\sum_{j\in R_i}\left[g(\hat{x}_i^{\,s},\boldsymbol{\theta})\right]\right) + \lambda\|\boldsymbol{\theta}\|$$
$$s.t. \quad g(\hat{x}_i^{\,s},\boldsymbol{\theta}) \geq 0 \;\; \forall\boldsymbol{\theta}, \mathbf{x}^s{}_i \qquad (33)$$
$$g(\hat{x}_i^{\,s}=0,\boldsymbol{\theta}) = 1 \quad \forall\boldsymbol{\theta}$$

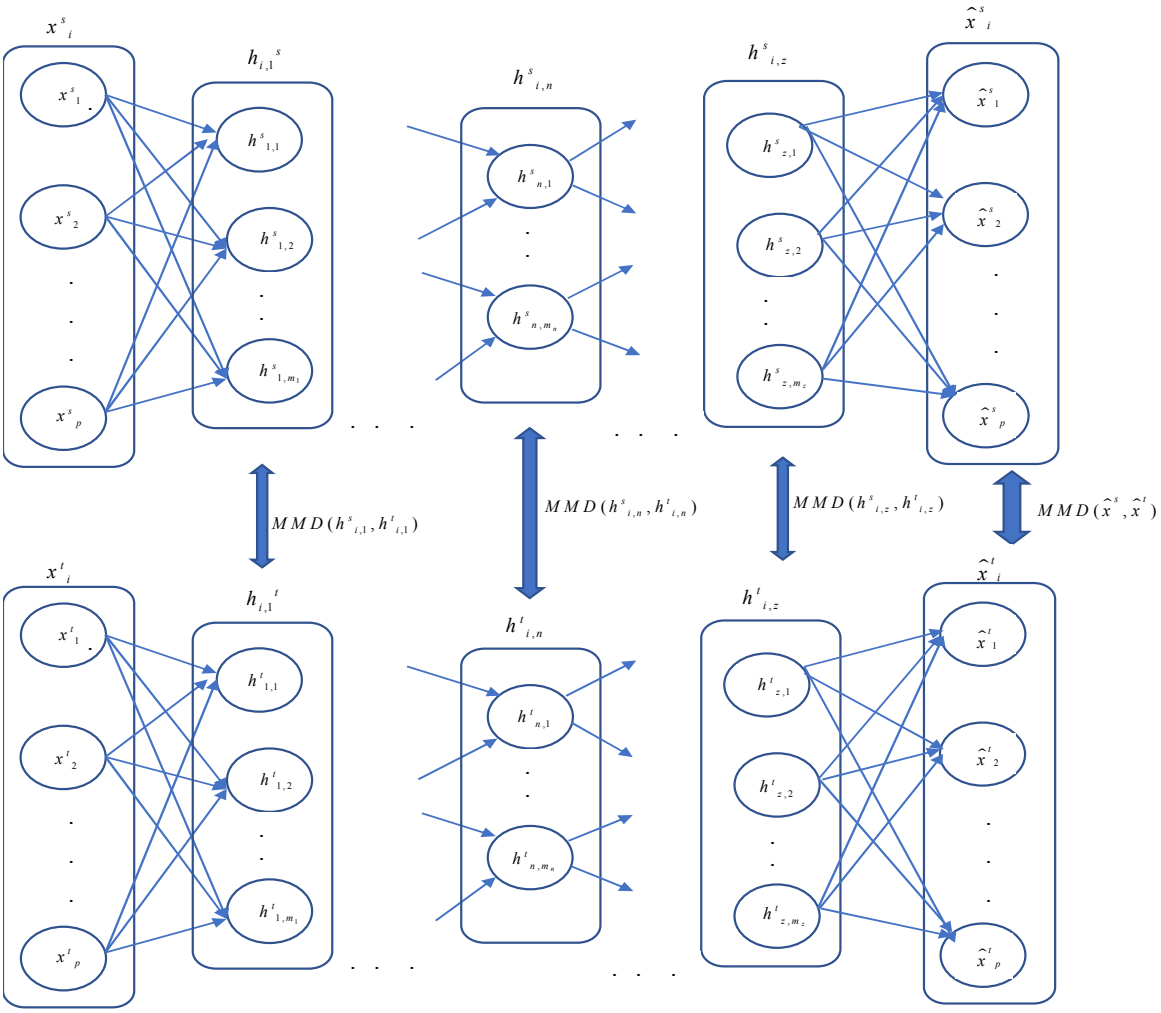

Figure 13. Structure of domain adoption

4.5 Simulation

In the simulation study we first generate two sets of samples with different distributions and failure times. Specifically, we generate samples consisting of two sets of covariates $\mathbf{x}^s_i, i = 1,...,100$ and $\mathbf{x}^t_j, j = 1,...,100$ which are drawn randomly from normal distributions with $\mathbf{x}^s_i \sim N(\mu^s = 1, \sigma^s = 2)$, $\mathbf{x}^t_i \sim N(\mu^t = 2, \sigma^t = 3)$. The covariates are chosen to have 4 elements, i.e. $\mathbf{x}^s_i, \mathbf{x}^t_j \in R^4$. Furthermore, $g(\mathbf{x} \mid \boldsymbol{\theta})$ is chosen to be a function which

satisfies the proposed properties in Chapter 3 [70] i.e., $g(x^s = 0, \boldsymbol{\theta}) = g(x_i^t = 0, \boldsymbol{\theta}) = 1$ and $g(x^s, \boldsymbol{\theta}), g(x^t, \boldsymbol{\theta}) \geq 0$. The functions are formulated as follows:

$$g(\mathbf{x}^s \mid \boldsymbol{\theta}) = g(\mathbf{x}^t \mid \boldsymbol{\theta}) = \begin{cases} 1 & \mathbf{x}^s = 0 \\ \left| \sum_{i=1}^{4} \sin(x_i) \right| & \mathbf{x}^s \neq 0 \end{cases} \tag{34}$$

where $|.|$ represents the absolute value and x_i represents the i^{th} component of a source and target sample. In this simulation study, we choose the Weibull lifetime distribution, which is commonly used in the reliability literature [55]. Based on the Weibull lifetime distribution, the simulated failure time and hazard function can be obtained using the inverse cumulative density probability function as follows:

$$T = \left(-\frac{\log(U)}{\lambda g(\mathbf{x}, \boldsymbol{\theta})} \right)^{1/\alpha}, h(t \mid \mathbf{x}) = \frac{d}{dt}\left(g(\mathbf{x} \mid \boldsymbol{\theta}) H_0(t) \right) = \lambda \alpha t^{\alpha - 1} g(\mathbf{x} \mid \boldsymbol{\theta}) \tag{35}$$

where $\lambda, \alpha > 0$ are the scalar and shape parameters of the Weibull distribution, respectively. We chose the Weibull distribution's parameter to be $\lambda = 0.5,\ \alpha = 0.5$.

Next, we apply the proposed model to estimate the reliability of the target domain. Specifically, the parameters of the model are estimated by 100 samples of a source's covariate with corresponding failure times and 100 samples of a target's covariates with corresponding failure times. We chose a single layer autoloader with two neurons. Also, the MLP with two hidden layers of two neurons are chosen.

Table 10 shows the values of a source and target covariates and their corresponding transferred values after applying the developed two-level autoencoder.

Table 10. Values of sources and target domains and their transferred values.

x^s	1.7284	2.5667	3.1544	-0.0080
x^t	0.9772	3.8489	-2.2507	2.5875
$\hat{x}^s$	0	0.0042`	0.4882	0
$\hat{x}^t$	0.9880	0	0.9896	0

To analyze the accuracy of the model, we compute the mean square error (MSE) [54] for the models of the 100 samples of the target's covariate. A smaller MSE indicates better performance. MSE is formulated as follows:

$$MSE = \frac{1}{N}\sum_{i}\left(g(\hat{\mathbf{x}}^t,\boldsymbol{\theta}) - \hat{g}(\hat{\mathbf{x}}^t,\boldsymbol{\theta})\right)^2 \tag{36}$$

where N is the test set size; $\hat{g}(\hat{\mathbf{x}}^t,\boldsymbol{\theta})$ is the estimated function of $g(\hat{\mathbf{x}}^t,\boldsymbol{\theta})$ for the i^{th} sample of target. Table 11 shows the MSE of the proposed model and the deep learning reliability model.

Table 11. Model performance

Method	**MSE**
Deep Learning reliability model	0.409
Transfer learning reliability model	0.164

In addition, the estimated parameters of MLP are given in Table 12. As the table shows, some of the parameters are zero.

Table 12. Model parameters

Parameters	Values	Parameters	Values
w_{11}^{1}	0.0752	w_{42}^{1}	2.9266
w_{21}^{1}	0	w_{52}^{1}	0
w_{31}^{1}	-3.1610	w_{11}^{2}	-0.0764
w_{41}^{1}	0.	w_{21}^{2}	0
w_{51}^{1}	0.1650	w_{12}^{2}	-5.3866
w_{12}^{1}	0	w_{22}^{2}	0
w_{22}^{1}	-0.5198	w_{11}^{O}	3.6536
w_{32}^{1}	0	w_{21}^{O}	5.0551

4.6 Case study

We applied our proposed model to DP steel. Specifically, for source domain, we obtained 20 microstructure images of a particular class of DP steel, DP780, with the corresponding failure time, $t_i^s, i = 1,..,20$, and 20 microstructure images of another class of DP steel, DP980, were chosen for the target domain. The size of each microstructure image is 100×100. Moreover, DP780 microstructure distribution is different from that of DP980. Figure 6 represents two samples of the DP780 and DP980 microstructures. The goal of the case chapter was to estimate the reliability of DP 980 by using the information of the source domain, (i.e. DP780). We first reduced the dimensions of the DP780 and

D980 microstructure images by applying autologistic regression to extract the covariates [57].

In this chapter, we applied autologistic regression with two neighboring orders to characterize the microstructures. The corresponding autologistic regression model considers 8 connecting neighbors $\{a_u, a_d, a_l, a_r, a_{ur}, a_{ul}, a_{dr}, a_{dl}\}$ [19]. Furthermore, we assume the DP AHSS materials are homogeneous and the microstructure images are anisotropic based on the assumption that $\{a_u, a_d\}, \{a_l, a_r\}, \{a_{ur}, a_{ul}\}$ and $\{a_{dr}, a_{dl}\}$ share their coefficients [19]. Hence, 5 covariates (coefficients) of each image were obtained including an intercept, i.e., $\mathbf{x}^s{}_i, \mathbf{x}^t{}_j \in R^5, i, j = 1,...,20.$ Next, we applied the proposed methodology to the dataset. We minimized the distribution distance of the DP780 and DP980 samples by transferring them to a new distribution space via the proposed two-level autoencoder. Table 13 shows the value of each covariate in a randomly selected microstructure image of source (DP780) and target (DP980) domains, i.e. $x^s{}_1, x^s{}_2, x^s{}_3, x^s{}_4$ and $x^t{}_1, x^t{}_2, x^t{}_3, x^t{}_4$, and their corresponding transferred covariate i.e. $\hat{x}^s{}_1, \hat{x}^s{}_2, \hat{x}^s{}_3, \hat{x}^s{}_4$ and $\hat{x}^t{}_1, \hat{x}^t{}_2, \hat{x}^t{}_3, \hat{x}^t{}_4$.

Table 13: source and target covariates with corresponding transferred covariates

$x^{s}{}_{0}$	-0.0799	$\widehat{x}^{s}{}_{0}$	0.9717
$x^{s}{}_{1}$	-0.0035	$\widehat{x}^{s}{}_{1}$	0.9548
$x^{s}{}_{2}$	-0.008	$\widehat{x}^{s}{}_{2}$	0.9946
$x^{s}{}_{3}$	-0.0051	$\widehat{x}^{s}{}_{3}$	0.9811
$x^{s}{}_{4}$	-0.0065	$\widehat{x}^{s}{}_{4}$	0.9299
$x^{t}{}_{0}$	-0.0613	$\widehat{x}^{t}{}_{0}$	0.9538
$x^{t}{}_{1}$	-0.0037	$\widehat{x}^{t}{}_{1}$	0.9819
$x^{t}{}_{2}$	-0.0072	$\widehat{x}^{t}{}_{2}$	0.9897
$x^{t}{}_{2}$	-0.0051	$\widehat{x}^{t}{}_{2}$	0.9933
$x^{t}{}_{4}$	-0.0073	$\widehat{x}^{t}{}_{4}$	0.9921

Next, the transferred covariates of DP780 used to train the developed MLP by minimizing the proposed loss function (33) to predict the hazard function of the DP980 samples.

The estimated model parameters of MLP are shown in Table 14.

Table 14. Model parameters

Parameters	Values	Parameters	Values
w_{11}^{1}	0.0010	w_{42}^{1}	0.0030
w_{21}^{1}	0.0160	w_{52}^{1}	-0.0030
w_{31}^{1}	0.0070	w_{11}^{2}	-0.0130
w_{41}^{1}	-0.0020	w_{21}^{2}	0.0070
w_{51}^{1}	0.0030	w_{12}^{2}	0
w_{12}^{1}	-0.0160	w_{22}^{2}	0.0020
w_{22}^{1}	-0.0040	w_{11}^{O}	0
w_{32}^{1}	0.0040	w_{21}^{O}	-0.0030

To further analyze and demonstrate the advantage of our proposed model, we compared the results of our model with existing models. We compared the model's performance with that of the deep learning-based reliability model and the Cox model. The parameters of the models were estimated by the source domain data, and then we applied the target's covariates to estimate the reliability. We used AIC to measure the model fitting. As it is shown in Table 15, our proposed model has superior performance over the deep learning reliability model.

Table 15. Comparing performance of models

Model	AIC
Cox Model	60.28
Deep Learning reliability model	62.20
Transfer learning reliability model	58.2

4.7 Conclusion

Reliability estimation is an important issue in the current competitive market. However, obtaining failure time for products or materials is a costly and time-consuming process, especially with today's high-quality products or materials. We propose a semi-parametric transfer learning-based reliability method to utilize the failure times and covariates of similar products or materials. Traditional reliability models are not efficient when the distribution of training and test covariates are not the same. To overcome these challenges, we propose two-level autoencoders to transfer the source and target domains to a new distribution space. The distribution discrepancy of the source and target domains' covariates in the new space is minimized. A novel loss function is developed to estimate the parameters of the two-level autoencoders. Moreover, a deep learning-based reliability model is used to capture the effect of the transferred covariates on failure time. Simulation studies and a case study were implemented to verify the proposed methodology and superior performance compared with the tradition model.

As future research topics, studies on a semi-supervised transfer learning-based reliability model and a transfer learning-based reliability model for time dependent covariates would be interesting.

Chapter 5. GENREAL CONCLUSION

Modern technologies generate covariates which can be utilized to improve failure time prediction. The covariates generally are high dimensional and topologically complex. We focus on incorporating the challenging covariates into reliability models. Although the failure time of advanced high strength steel is chosen to illustrate the proposed model and develop methods, the methodologies may be applicable to other materials or products.

Studies show that microstructure strongly affects a material's physical properties including failure time. Without incorporating the microstructure as covariates, the reliability estimation of the materials may not be accurate.

In this dissertation, we focus on incorporating a microstructure to improve the reliability prediction accuracy of materials by developing a statistical approach to model the complex structure of a microstructure, a deep learning reliability model to capture the complex effect of the microstructures on the failure time of the materials, and a transfer learning model to utilize the failure time of a type of covariate to predict the reliability of another type of covariate.

In CHAPTER 2, we propose a statistical method to reduce the dimension and complexity of a covariate. The proposed model considers the spatial properties of covariates, but unlike traditional dimension reduction models, orientation does not affect the performance of the model. Specifically, we propose a penalized autologistic regression model that includes the parameter selection process by removing the redundant or irrelevant model parameters (or neighbors). As a result, a high order of neighboring can be applied to model complex covariates. The model selection process

that selects the most appropriate parameters (or neighbor structure) can be automatically implemented during the model parameter estimation process. A maximum likelihood estimation method can be developed to estimate the model parameters. However, the likelihood function has a complex form which makes the parameter estimation time consuming. We develop a penalized pseudo log-likelihood function to tackle the challenge. When the size of the sample is large, applying classical optimization methods to maximize the penalized pseudo log-likelihood function still takes much time. To overcome this difficulty, we developed a new approximated accelerated proximal gradient method. The developed methodologies are verified and demonstrated through designed physical experiments. The methods are also applicable to all binary images to extract certain patterns.

In CHAPTER 3, we propose a novel deep learning-based reliability model by considering complex covariates. The proposed model is a semi-parametric and proportional hazard model. The model captures the complex relationship between covariates and failure time, unlike traditional reliability models which are inefficient due to their underlying assumptions.

To estimate model parameters, since the baseline hazard function is not defined, MLE cannot be applied directly. Moreover, the traditional OLS loss function of MLP is not applicable, as there is no access to the true values of the output during the training process. To overcome the challenge, a novel method based on the partial likelihood framework is developed. Furthermore, when there are few samples available, which generally appears in reliability problems, the method may suffer from overfitting. To overcome this challenge, we develop a loss function based on penalized partial likelihood.

The developed model can overcome the challenge of right censored failure data. The aforementioned methods are illustrated using both simulation studies and designed physical experiments on advanced high strength steel. Results shows improvement in reliability prediction compared to traditional reliability models

In CHAPTER 4, we develop a transfer learning-based reliability model. The proposed model predicts the reliability of a subject by utilizing the covariate and failure time of similar subjects. Specifically, the proposed model consists of a two-level autoencoder to minimize the distribution of covariates of subjects of interest and similar subjects to improve the performance of the model.

To estimate the e parameters of the model, we develop a novel loss function for the two-level autoencoder. Furthermore, the MMD statistic is used to minimize the distribution of each layer. A simulation study is conducted to verify the developed methods. Moreover, physical experiments on advanced high strength steel are conducted to demonstrate the proposed model. Results show that accuracy of failure time predictions are improved by the proposed transfer learning-based reliability method compare with traditional reliability models.

Appendix A. Proof of Proposition 1

Assume that the given microstructure has n pixels, and we consider a model with the p order of neighboring system. $f(\boldsymbol{\lambda})$ in (7) is a Lipschitz continuous gradient. If (6) holds for a constant *L* and $\boldsymbol{\alpha}=[\alpha_0,..\alpha_p],\boldsymbol{\theta}=[\theta_0,...\theta_p]\in\square^{p+1}$. Additionally, for notation simplification, we define $z_{i,j}\in\{0,1\},j\in N(i)$ to represent the j^{th} neighbor of site *i*. So $f(\boldsymbol{\lambda})$ in (7) can be rewrite as follows

$$f(\boldsymbol{\lambda})=-\sum_{i=1}^{n}\log\left(\frac{\exp(x_i\sum_{j=0}^{p}\lambda_j z_{i,j})}{1+\exp(\sum_{j=0}^{p}\lambda_j z_{i,j})}\right) \tag{37}$$

where $z_{i,0}=1$. The gradient of $f(\boldsymbol{\lambda})$ is defined as $\nabla f(\boldsymbol{\lambda})=<\frac{\partial f}{\partial\lambda_0},...,\frac{\partial f}{\partial\lambda_k}>$. The partial differential can be expressed as follows:

$$\frac{\partial f}{\partial\lambda_j}=-x_1z_{1,j}+\frac{z_{1,j}\times\exp(\sum_{j=0}^{p}\lambda_j z_{1,j})}{1+\exp(\sum_{j=0}^{p}\lambda_j z_{1,j})}+...-x_nz_{n,j}+\frac{z_{n,j}\times\exp(\sum_{j=0}^{p}\lambda_j z_{n,j})}{1+\exp(\sum_{j=0}^{p}\lambda_j z_{n,j})} \tag{38}$$

By inserting (38) into (6), we obtain the following equation:

$$\| \nabla f(\boldsymbol{\alpha}) - \nabla f(\boldsymbol{\theta}) \|^2 = [-x_1 z_{1,0} + \frac{z_{1,0} \times \exp(\sum_{j=0}^{p} \alpha_j z_{1,j})}{1+\exp(\sum_{j=0}^{p} \alpha_j z_{1,j})} + \ldots - x_n z_{n,0} + \frac{z_{n,0} \times \exp(\sum_{j=0}^{p} \alpha_j z_{n,j})}{1+\exp(\sum_{j=0}^{p} \alpha_j z_{n,j})} -$$

$$(-x_1 z_{1,0} + \frac{z_{1,0} \times \exp(\sum_{j=0}^{p} \theta_j z_{1,j})}{1+\exp(\sum_{j=0}^{p} \theta_j z_{1,j})} + \ldots - x_n z_{n,0} + \frac{z_{n,0} \times \exp(\sum_{j=0}^{p} \theta_j z_{n,j})}{1+\exp(\sum_{j=0}^{p} \theta_j z_{n,j})})]^2 + \ldots.$$

$$+[-x_1 z_{1,k} + \frac{z_{1,k} \times \exp(\sum_{j=0}^{p} \alpha_j z_{1,j})}{1+\exp(\sum_{j=0}^{p} \alpha_j z_{1,j})} + \ldots - x_n z_{n,k} + \frac{z_{n,k} \times \exp(\sum_{j=0}^{p} \alpha_j z_{n,j})}{1+\exp(\sum_{j=0}^{p} \alpha_j z_{n,j})} -$$

$$(-x_1 z_{1,k} + \frac{z_{1,k} \times \exp(\sum_{j=0}^{p} \theta_j z_{1,j})}{1+\exp(\sum_{j=0}^{p} \theta_j z_{1,j})} + \ldots - x_n z_{n,k} + \frac{z_{n,k} \times \exp(\sum_{j=0}^{p} \theta_j z_{n,j})}{1+\exp(\sum_{j=0}^{p} \theta_j z_{n,j})})]^2$$

(39)

As $z_{i,j}$ is a binary variable, the maximum of (39) happens when $z_{i,j} = 1$, $i \in \{1, 2, \ldots, n\}$, $j \in \{0, 1, \ldots, p\}$. By replacing all $z_{i,j}$ with 1 and x_i with $c > 1$, the maximum value of (39) is represented as follows.

$$\| \nabla f(\boldsymbol{\alpha}) - \nabla f(\boldsymbol{\theta}) \|^2 \leq \left[\begin{array}{c} \frac{\exp(\sum_{j=0}^{p} \alpha_j)}{1+\exp(\sum_{j=0}^{p} \alpha_j x_j^{\,1})} + \ldots + \frac{\exp(\sum_{j=0}^{p} \alpha_j)}{1+\exp(\sum_{j=0}^{p} \alpha_j)} - \\ \frac{\exp(\sum_{j=0}^{p} \theta_j)}{1+\exp(\sum_{j=0}^{p} \theta_j)} \ldots - \frac{\exp(\sum_{j=0}^{p} \theta_j)}{1+\exp(\sum_{j=0}^{p} \theta_j)}) \end{array} \right]^2 \ldots.$$

$$+ \left[\begin{array}{c} \frac{\exp(\sum_{j=0}^{p} \alpha_j)}{1+\exp(\sum_{j=0}^{p} \alpha_j)} + \ldots + \frac{\exp(\sum_{j=0}^{p} \alpha_j)}{1+\exp(\sum_{j=0}^{p} \alpha_j)} - \\ \frac{\exp(\sum_{j=0}^{p} \theta_j)}{1+\exp(\sum_{j=0}^{p} \theta_j)} \ldots - \frac{\exp(\sum_{j=0}^{p} \theta_j)}{1+\exp(\sum_{j=0}^{p} \theta_j)} \end{array} \right]^2 = c^2 (p+1) n \times n \left(\frac{\exp(\sum_{j=0}^{p} \alpha_j)}{1+\exp(\sum_{j=0}^{p} \alpha_j)} - \frac{\exp(\sum_{j=0}^{p} \theta_j)}{1+\exp(\sum_{j=0}^{p} \theta_j)} \right)^2$$

(40)

For $c>1$. Based on (40) and the definition of Lipschitz continuous, in order to prove Proposition 1 the following inequality has to hold

$$c^2 (p+1) n \times n \left(\frac{\exp(\sum_{j=0}^{p} \alpha_j)}{1+\exp(\sum_{j=0}^{p} \alpha_j)} - \frac{\exp(\sum_{j=0}^{p} \theta_j)}{1+\exp(\sum_{j=0}^{p} \theta_j)} \right)^2 \leq c^2 (p+1) n^2 \| \boldsymbol{\alpha} - \boldsymbol{\theta} \|^2 \qquad (41)$$

Since $0 \leq \frac{\exp(\sum_{j=0}^{p} \alpha_j)}{1+\exp(\sum_{j=0}^{p} \alpha_j)} \leq 1$ and $0 \leq \frac{\exp(\sum_{j=0}^{p} \theta_j)}{1+\exp(\sum_{j=0}^{p} \theta_j)} \leq 1$, therefore:

$$-1 \le \frac{\exp(\sum_{j=0}^{p} \alpha_j)}{1+\exp(\sum_{j=0}^{p} \alpha_j)} - \frac{\exp(\sum_{j=0}^{p} \theta_j)}{1+\exp(\sum_{j=0}^{p} \theta_j)} \le 1 \tag{42}$$

Since $\| \boldsymbol{\alpha} - \boldsymbol{\theta} \| \ge 0$, to prove Proposition 1, it is sufficient to show that for those values of $\boldsymbol{\alpha} = [\alpha_0, ..\alpha_p], \boldsymbol{\theta} = [\theta_0, ...\theta_p] \in \square^{p+1}$, (42) is non-negative and the following holds:

$$0 \le \frac{\exp(\sum_{j=0}^{p} \alpha_j)}{1+\exp(\sum_{j=0}^{p} \alpha_j)} - \frac{\exp(\sum_{j=0}^{p} \theta_j)}{1+\exp(\sum_{j=0}^{p} \theta_j)} \le \| \boldsymbol{\alpha} - \boldsymbol{\theta} \| \le 1 \tag{43}$$

$\| \boldsymbol{\alpha} - \boldsymbol{\theta} \| \le 1$ if and only if $|\alpha_i - \theta_i| \le 1$ for all $i \in \{0, ..., p\}$. Assume that $|\alpha_i - \theta_i| = \varepsilon_i \le 1$ for all $i \in \{0, ..., p\}$ and $\varepsilon = \min\{\varepsilon_i\}$.

$$\| \boldsymbol{\alpha} - \boldsymbol{\theta} \| = \sqrt{\sum_{i=0}^{p} \varepsilon_i} \ge \sqrt{(p+1)\varepsilon} = \| \boldsymbol{\alpha}' - \boldsymbol{\theta}' \| \tag{44}$$

where $\boldsymbol{\alpha}' = \boldsymbol{\alpha}, \boldsymbol{\theta}' = \{\alpha_0 + \varepsilon, ..., \alpha_p + \varepsilon\}$

Based on (43) and (44), the following inequality needs to be proved.

$$\frac{1}{1+\exp(\sum_{j=0}^{p} \alpha_j)} - \frac{1}{1+\exp(\sum_{j=0}^{p} \alpha_j + \varepsilon)} \le \sqrt{(p+1)\varepsilon} \tag{45}$$

Note that we know that the left-hand side is always non-negative. We can rewrite the left hand side of (45) as follows

$$\frac{1}{1+\exp(\sum_{j=0}^{p}\alpha_j)} - \frac{1}{1+\exp((p+1)\varepsilon)\times\exp(\sum_{j=0}^{p}\alpha_j)} \le \sqrt{(p+1)\varepsilon} \tag{46}$$

By considering $x = \exp(\sum_{j=0}^{p}\alpha_j),\ y = (p+1)\varepsilon$, we can rewrite (46) as follows:

$$z(x,y) = \frac{1}{1+x} - \frac{1}{1+\exp(y)\times x} \le \sqrt{y} \tag{47}$$

where $z(x,y)$ is a continuous and non-decreasing function for fixed x and the desired domain $0 \le y \le 1$. Furthermore, $z(x,y)$ has maximum values when $0 \le x \le 1$ and $0 \le y \le 1$; however, for those values $\sqrt{(p+1)\varepsilon}$ is much larger than $z(x,y)$.

Appendix B. Proof of Proposition 2

The first Karush–Kuhn–Tucker (KKT) condition of (8) can be expressed as follows

$$\nabla f(\boldsymbol{\lambda}^{m-1}) + \nabla\left(<\boldsymbol{\lambda}-\boldsymbol{\lambda}^{m-1}, \nabla(f(\boldsymbol{\lambda}^{m-1})>\right) + \nabla\left(\frac{L}{2}\|\boldsymbol{\lambda}-\boldsymbol{\lambda}^{m-1}\|^2\right) + \beta\mathbf{g} = 0 \tag{48}$$

where $\|.\|$ is a L2-norm, $<.>$ is inner product, and L is the constant obtained from Proposition 1.

Based on the notation defined in Proposition 1, we first evaluate each term in (48). Since $f(\boldsymbol{\lambda}^{m-1})$ is not dependent on $\boldsymbol{\lambda}$, $\nabla f(\boldsymbol{\lambda}^{m-1}) = 0$. The next term in (48) is $<\boldsymbol{\lambda}-\boldsymbol{\lambda}^{m-1}, \nabla(f(\boldsymbol{\lambda}^{m-1})>$ that can be expressed as follows using (38)

$$< \boldsymbol{\lambda} - \boldsymbol{\lambda}^{m-1}, \nabla(f(\boldsymbol{\lambda}^{m-1}) >= \sum_{t=0}^{p} \left[\begin{array}{l} (\lambda_t - \lambda_t^{m-1}) \times (-x_1 z_{1,t} + \dfrac{z_{1,t} \times \exp(\sum_{j=0}^{p} \lambda^{m-1}{}_j z_{1,j})}{1 + \exp(\sum_{j=0}^{p} \lambda^{m-1}{}_j z_{1,j})} + \\ \ldots - x_n z_{n,t} + \dfrac{z_{n,t} \times \exp(\sum_{j=0}^{p} \lambda^{m-1}{}_j z_{n,j})}{1 + \exp(\sum_{j=0}^{p} \lambda^{m-1}{}_j z_{n,j})}) \end{array} \right] \tag{49}$$

where $z_{i,0} = 1,\ i = 1,\ldots,n$. The derivative of (49) is:

$$\begin{aligned} &\frac{\partial\left(< \boldsymbol{\lambda} - \boldsymbol{\lambda}^{m-1}, \nabla(f(\boldsymbol{\lambda}^{m-1}) >\right)}{\partial(\lambda_t)} = -x_1 z_{1,t} + \frac{z_{1,t} \times \exp(\sum_{j=0}^{p} \lambda^{m-1}{}_j z_{1,j})}{1 + \exp(\sum_{j=0}^{p} \lambda^{m-1}{}_j z_{1,j})} + \ldots - x_n z_{n,t} \\ &+ \frac{z_{n,t} \times \exp(\sum_{j=0}^{p} \lambda^{m-1}{}_j z_{n,j})}{1 + \exp(\sum_{j=0}^{p} \lambda^{m-1}{}_j z_{n,j})} \\ &t = 0 \ldots p \end{aligned} \tag{50}$$

$\frac{L}{2} \| \boldsymbol{\lambda} - \boldsymbol{\lambda}^{m-1} \|^2$ in (48) can be expressed as follows

$$\begin{aligned} &\frac{L}{2} \| \boldsymbol{\lambda}^{m-1} - \boldsymbol{\lambda} \|^2 = \frac{L}{2} (\boldsymbol{\lambda}^{m-1} - \mathbf{I}\boldsymbol{\lambda})^T (\boldsymbol{\lambda}^{m-1} - \mathbf{I}\boldsymbol{\lambda}) = \\ &= \frac{L}{2} (\boldsymbol{\lambda}^{m-1})^T \boldsymbol{\lambda}^{m-1} - \frac{L}{2} (\boldsymbol{\lambda}^T \boldsymbol{\lambda}^{m-1} + (\boldsymbol{\lambda}^{m-1})^T \boldsymbol{\lambda}) + L \| \boldsymbol{\lambda} \|^2 \end{aligned} \tag{51}$$

where $\mathbf{I}$ is the identity matrix. In vector space given $A \in R^{k+1}$, $B \in R^{k+1}$, then $A^T \times B = B^T \times A$ where A^T, B^T represent transpose of A, B, respectively. Using the matrix transpose property, we rewrite (51) as follows:

$$\frac{L}{2} \| \boldsymbol{\lambda}^{m-1} - \boldsymbol{\lambda} \|^2 = \frac{L}{2} (\boldsymbol{\lambda}^{m-1})^T \boldsymbol{\lambda}^{m-1} - L(\sum_{j=0}^{p} \lambda_j \lambda_j^{m-1}) + L \sum_{j=0}^{p} \lambda_j{}^2 \tag{52}$$

Furthermore, the derivative of the above term can be expressed as follows

$$\frac{\partial\left(\frac{L}{2}\| \lambda_j^{m-1} - \lambda_j \|^2\right)}{\partial(\lambda_j)} + \beta g_j = -L\lambda_j^{m-1} + 2L\lambda_j + \beta g_j = 0,\ g_j = \begin{cases} [-1,1] & \lambda_j = 0 \\ sgn(\lambda_j) & \lambda_j \neq 0 \end{cases} \tag{53}$$

With approximation of $1+\exp(\sum_{j=0}^{p}\lambda^{m-1}{}_j z_{i,j}) \approx \exp(\sum_{j=0}^{p}\lambda^{m-1}{}_j z_{i,j})$ in (50) and combine (50) and (53), we have

$$G(\lambda_j) = \frac{\partial\left(<\boldsymbol{\lambda} - \boldsymbol{\lambda}^{m-1}, \nabla(f(\boldsymbol{\lambda}^{m-1})>\right)}{\partial(\lambda_t)} - \frac{\partial\left(\frac{L}{2}\| \lambda_j^{m-1} - \lambda_j \|^2\right)}{\partial(\lambda_j)} + \beta g_j = -x_1 z_{1,j} + z_{1,j} + \ldots - x_n z_{n,j}$$
$$+ z_{n,j} - L\lambda_j^{m-1} + 2L\lambda_j + \beta g_j = 0 \tag{54}$$

Note that the approximation $1+\exp(\sum_{j=0}^{p}\lambda^{m-1}{}_j z_{i,j}) \approx \exp(\sum_{j=0}^{p}\lambda^{m-1}{}_j z_{i,j})$ is always true since the model has many parameters and some of the model parameters are zero which makes $\exp(\sum_{j=0}^{p}\lambda^{m-1}{}_j z_{i,j}) \gg 1$.

Moreover, (54) can be reformulated as

$$\lambda_j = \frac{x_1 z_{1,j} - z_{1,j} \ldots + x_n z_{n,j} - z_{n,j}}{2L} - \frac{\beta \times g_j}{2L} + \frac{(\lambda_j^{m-1})}{2} \tag{55}$$

We consider two cases for the solution. First, if $\lambda_j = 0$ then

$$x_1 z_{1,j} - z_{1,j} \ldots + x_n z_{n,j} + z_{n,j} + L(\lambda_j^{m-1}) = \beta g_j \in [-\beta, +\beta] \tag{56}$$

Second, if $\lambda_j \neq 0$ then

$$\lambda_j + \frac{\beta \times \mathrm{sgn}(\lambda_j)}{2L} = \frac{x_1 z_{1,j} - z_{1,j} \ldots + x_n z_{n,j} - z_{n,j}}{2L} + \frac{(\lambda_j^{m-1})}{2} \tag{57}$$

The final solution can be in the compact form of:

$$\lambda_j = \mathrm{sgn}\left(\frac{x_1 z_{1,j} - z_{1,j} \ldots + x_n z_{n,j} - z_{n,j}}{2L} + \frac{(\lambda_j^{m-1})}{2}\right)\left(\left|\frac{x_1 z_{1,j} - z_{1,j} \ldots + x_n z_{n,j} - z_{n,j}}{2L} + \frac{(\lambda_j^{m-1})}{2}\right| - \frac{\beta}{2L}\right)_+$$

where $\mathrm{sgn}(.)$ is a sign function and $x_+ = \max(0, x)$.

Appendix C. Proof of convergence of Algorithm 1

Let $G'(\lambda_j)$ represents solution of (48) with the approximate and $G'(\lambda^*_j)$ represent the solution without approximate. In our case, the following inequality always holds:

$$G'(\lambda^*_j) - G'(\lambda_j) \leq -n \tag{58}$$

By replacing (58) with their corresponding functions, we have the following inequality

$$\lambda_j^* \leq \lambda_j + \frac{n}{2L} \tag{59}$$

where λ_j^*, λ_j is the parameter estimation corresponding with $G'(\lambda^*_j), G'(\lambda_j)$, respectively. Furthermore, let $G(\lambda)$ represents sum of $f(\lambda)$ and $g(\lambda)$, i.e., $G(\boldsymbol{\lambda}) = f(\boldsymbol{\lambda}) + g(\boldsymbol{\lambda})$. By plugging (59) into $G(\lambda)$, we have the following

$$G(\boldsymbol{\lambda}^{*(m)}) \leq G(\boldsymbol{\lambda}^m) + \beta\frac{(p+1)n}{2L} + \frac{n}{L} + c \tag{60}$$

where $c \geq 1$ is a constant. [71] showed the following inequality holds

$$G(\boldsymbol{\lambda}^{*(m)}) - G^+ \leq \frac{2}{(M+1)^2 L} \| \boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^+ \|_2^2 \tag{61}$$

where $G^+ = \min_{\lambda} G(\boldsymbol{\lambda})$, $\boldsymbol{\lambda}^+ = \arg\min_{\lambda} G(\boldsymbol{\lambda})$ and M is the maximum iteration. By plugging (60) into (61), we have the following inequality

$$G(\boldsymbol{\lambda}^m)-G^+ \le \frac{2}{(M+1)^2 L}\| \boldsymbol{\lambda}^0-\boldsymbol{\lambda}^+ \|_2^2 -(\beta\frac{(p+1)n}{2L}+\frac{n}{L}+c) \quad (62)$$

The last inequality shows that $G(\boldsymbol{\lambda}^m)-G^+$ decreases as fast as $O(\frac{1}{M^2})$.

Appendix D. Algorithm for reconstruction of a microstructure

Algorithm 3:

Given a fraction parameter f, the model parameters $\{\lambda_0,\lambda_2,...,\lambda_p\}$, initial temperature T_1, cooling rate α, maximum iteration t_n, and the size of sample $d\times d$

1. Initialize the lattice $\mathbf{X}$ according to f
2. For i in $1:t_n$
3. Randomly select two pixels X_l and X_j
4. Assign $\mathbf{X}^{'}=\mathbf{X}$
5. $\mathbf{X}^{'}=$ Switch value of X_l, X_j in $\mathbf{X}^{'}$
6. If $l_{PPLL}(\mathbf{X})<l_{PPLL}(\mathbf{X}^{'})$ then $\mathbf{X}=\mathbf{X}^{'}$

   Else set $\mathbf{X}=\mathbf{X}^{'}$ with probability of $\exp\left(-[l_{PPLL}(\mathbf{X})-l_{PPLL}(\mathbf{X}^{'})]/T_i\right)$
7. $T_{i+1}=\alpha T_i$ and go to step 2

Appendix E. Proof of Proposition 3:

If random variable time-to-failure T has survival function $S(t)$, then the transfer random variable follows uniform distribution on $[0,1]$ [72]

$$S(T) \sim Uniform[0,1] \tag{63}$$

Consequently, the cumulative hazard function follows exponential function with model parameter $\mu = 1$ [72]

$$H(T) = -\log S(T) \sim E\,\mathrm{xp}(1) \tag{64}$$

Furthermore, the cumulative hazard function of the proposed model in (12) is calculated as follow

$$\begin{aligned} \hat{H}(t) &= -\log(\hat{S}(t)) = -\log\left(\hat{S}_0(t)^{g(\mathbf{x}|\boldsymbol{\theta})}\right) = -g(\mathbf{x}\,|\,\boldsymbol{\theta})\log(\hat{S}_0(t)) \\ &= g(\mathbf{x}\,|\,\boldsymbol{\theta})\int_0^t \hat{h}_0(s)ds = g(\mathbf{x}\,|\,\boldsymbol{\theta})\hat{H}_0(t) \end{aligned} \tag{65}$$

Where $\hat{H}_0(t)$ is the Breslow estimator introduced in [73].

REFRENCES

1. Blair, A., et al., *Reliability of reporting on life-style and agricultural factors by a sample of participants in the Agricultural Health Study from Iowa.* Epidemiology, 2002. **13**(1): p. 94-99.

2. Wei, L.-J., *The accelerated failure time model: a useful alternative to the Cox regression model in survival analysis.* Statistics in medicine, 1992. **11**(14-15): p. 1871-1879.

3. Ryu, C., et al., *Microstructure and reliability of copper interconnects.* IEEE transactions on electron devices, 1999. **46**(6): p. 1113-1120.

4. Kuziak, R., R. Kawalla, and S. Waengler, *Advanced high strength steels for automotive industry.* Archives of civil and mechanical engineering, 2008. **8**(2): p. 103-117.

5. Alpas, A. and J. Zhang, *Effect of microstructure (particulate size and volume fraction) and counterface material on the sliding wear resistance of particulate-reinforced aluminum matrix composites.* Metallurgical and Materials Transactions A, 1994. **25**(5): p. 969-983.

6. Zaefferer, S., J. Ohlert, and W. Bleck, *A study of microstructure, transformation mechanisms and correlation between microstructure and mechanical properties of a low alloyed TRIP steel.* Acta Materialia, 2004. **52**(9): p. 2765-2778.

7. Paul, S.K., *Real microstructure based micromechanical model to simulate microstructural level deformation behavior and failure initiation in DP 590 steel.* Materials & Design, 2013. **44**: p. 397-406.

8. Jiao, Y., F. Stillinger, and S. Torquato, *Modeling heterogeneous materials via two-point correlation functions: Basic principles.* Physical Review E, 2007. **76**(3): p. 031110.

9. Saheli, G., H. Garmestani, and A. Gokhale, *Homogenization relations for elastic properties of two-phase composites using two-point statistical functions.* Journal of Mechanics of Materials and Structures, 2008. **3**(1): p. 85-106.

10. Torquato, S., *Random heterogeneous materials: microstructure and macroscopic properties*. Vol. 16. 2013: Springer-Verlag New York.

11. Zhang, N., Q. Yang, and X. Wu, *A Semi-parametric Model for Microstructure Analysis of Advanced High-strength Dual-phase Steels Considering Sample Variation.* Quality and Reliability Engineering International, 2016. **32**(8): p. 2777-2788.

12. Feng, J., et al., *Statistical reconstruction of two-phase random media.* Computers & Structures, 2014. **137**: p. 78-92.

13. Huffer, F.W. and H. Wu, *Markov chain Monte Carlo for autologistic regression models with application to the distribution of plant species.* Biometrics, 1998. **54**: p. 509-524.

14. Besag, J., *Spatial interaction and the statistical analysis of lattice systems.* Journal of the Royal Statistical Society. Series B (Methodological), 1974. **36**: p. 192-236.

15. Augustin, N., M. Mugglestone, and S. Buckland, *An autologistic model for the spatial distribution of wildlife.* Journal of Applied Ecology, 1996. **33**: p. 339-347.

16. He, F., J. Zhou, and H. Zhu, *Autologistic regression model for the distribution of vegetation.* Journal of agricultural, biological, and environmental statistics, 2003. **8**(2): p. 205.

17. Arbia, G., R. Benedetti, and G. Espa, *Contextual classification in image analysis: an assessment of accuracy of ICM.* Computational statistics & data analysis, 1999. **30**(4): p. 443-455.

18. Cross, G.R. and A.K. Jain, *Markov random field texture models.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1983. **PAMI-5**(1): p. 25-39.

19. Zhang, N. and Q. Yang, *A random effect autologistic regression model with application to the characterization of multiple microstructure samples.* IIE Transactions, 2016. **48**(1): p. 34-42.

20. Attardi, L., M. Guida, and G. Pulcini, *A mixed-Weibull regression model for the analysis of automotive warranty data.* Reliability Engineering & System Safety, 2005. **87**(2): p. 265-273.

21. Bennett, S., *Log-logistic regression models for survival data.* Applied Statistics, 1983: p. 165-171.

22. Si, W., Q. Yang, and X. Wu, *A distribution-based functional linear model for reliability analysis of advanced high-strength dual-phase steels by utilizing material microstructure images.* IISE Transactions, 2017. **49**(9): p. 863-873.

23. Cox, D.R., *Regression models and life-tables*, in *Breakthroughs in statistics*. 1992, Springer. p. 527-541.

24. Harrell, F.E., *Cox proportional hazards regression model*, in *Regression modeling strategies*. 2015, Springer. p. 475-519.

25. Zhang, H.H. and W. Lu, *Adaptive Lasso for Cox's proportional hazards model.* Biometrika, 2007. **94**(3): p. 691-703.

26. Tibshirani, R., *The lasso method for variable selection in the Cox model.* Statistics in medicine, 1997. **16**(4): p. 385-395.

27. Sleeper, L.A. and D.P. Harrington, *Regression splines in the Cox model with application to covariate effects in liver disease.* Journal of the American Statistical Association, 1990. **85**(412): p. 941-949.

28. Faraggi, D. and R. Simon, *A neural network model for survival data.* Statistics in medicine, 1995. **14**(1): p. 73-82.

29. Pijnenburg, M., *Additive hazards models in repairable systems reliability.* Reliability Engineering & System Safety, 1991. **31**(3): p. 369-390.

30. Badıa, F., M.D. Berrade, and C.A. Campos, *Aging properties of the additive and proportional hazard mixing models.* Reliability Engineering & System Safety, 2002. **78**(2): p. 165-172.

31. Kalinin, S.V., B.G. Sumpter, and R.K. Archibald, *Big–deep–smart data in imaging for guiding materials design.* Nature materials, 2015. **14**(10): p. 973.

32. Sherman, M., T.V. Apanasovich, and R.J. Carroll, *On estimation in binary autologistic spatial models.* Journal of Statistical Computation and Simulation, 2006. **76**(2): p. 167-179.

33. Gumpertz, M.L., J.M. Graham, and J.B. Ristaino, *Autologistic model of spatial pattern of Phytophthora epidemic in bell pepper: effects of soil variables on disease presence.* Journal of Agricultural, Biological, and Environmental Statistics, 1997. **2**: p. 131-156.

34. Tibshirani, R., *Regression shrinkage and selection via the lasso.* Journal of the Royal Statistical Society. Series B (Methodological), 1996. **58**: p. 267-288.

35. Lee, S.-I., et al. *Efficient l~ 1 regularized logistic regression*. in *AAAI*. 2006.

36. Parikh, N. and S. Boyd, *Proximal algorithms.* Foundations and Trends® in Optimization, 2014. **1**(3): p. 127-239.

37. Stone, M., *Cross-validatory choice and assessment of statistical predictions.* Journal of the Royal Statistical Society. Series B (Methodological), 1974. **36**: p. 111-147.

38. Chai, T. and R.R. Draxler, *Root mean square error (RMSE) or mean absolute error (MAE)?* Geoscientific Model Development Discussions, 2014. **7**: p. 1525-1534.

39. Irani, M. and A.K. Taheri, *Effect of forging temperature on homogeneity of microstructure and hardness of precision forged steel spur gear.* Materials Chemistry and Physics, 2008. **112**(3): p. 1099-1105.

40. Varma, M. and A. Zisserman, *A statistical approach to texture classification from single images.* International Journal of Computer Vision, 2005. **62**(1): p. 61-81.

41. Kanungo, T., et al., *An efficient k-means clustering algorithm: Analysis and implementation.* IEEE transactions on pattern analysis and machine intelligence, 2002. **24**(7): p. 881-892.

42. Nunes, J.C., et al. *Texture analysis based on the bidimensional empirical mode decomposition with gray-level co-occurrence models*. in *Signal Processing and Its Applications, 2003. Proceedings. Seventh International Symposium on*. 2003. IEEE.

43. Bruzzone, L., S. Serpico, and G. Vernazza. *Effects of parameter tuning and de-speckle filtering on the accuracy of SAR image classification based on gray-level co-occurrence matrix features*. in *Geoscience and Remote Sensing, 1997. IGARSS'97. Remote Sensing-A Scientific Vision for Sustainable Development., 1997 IEEE International*. 1997. IEEE.

44. Bishop, C.M., *Neural networks for pattern recognition*. 1995: Oxford university press.

45. Wan, E.A., *Neural network classification: A Bayesian interpretation.* IEEE Transactions on Neural Networks, 1990. **1**(4): p. 303-305.

46. Specht, D.F., *A general regression neural network.* IEEE transactions on neural networks, 1991. **2**(6): p. 568-576.

47. Glorot, X., A. Bordes, and Y. Bengio. *Deep sparse rectifier neural networks*. in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. 2011.

48. *Mean Absolute Error*, in *Encyclopedia of Machine Learning*, C. Sammut and G.I. Webb, Editors. 2010, Springer US: Boston, MA. p. 652-652.

49. Cox, D.R. and E.J. Snell, *A general definition of residuals.* Journal of the Royal Statistical Society. Series B (Methodological), 1968: p. 248-275.

50. Massey Jr, F.J., *The Kolmogorov-Smirnov test for goodness of fit.* Journal of the American statistical Association, 1951. **46**(253): p. 68-78.

51. Cox, D.R., *Partial likelihood.* Biometrika, 1975. **62**(2): p. 269-276.

52. Panchal, G. and M. Panchal, *Review on methods of selecting number of hidden nodes in artificial neural network.* International Journal of Computer Science and Mobile Computing, 2014. **3**(11): p. 455-464.

53. Stone, M., *Cross-validatory choice and assessment of statistical predictions.* Journal of the royal statistical society. Series B (Methodological), 1974: p. 111-147.

54. Willmott, C.J. and K. Matsuura, *Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance.* Climate research, 2005. **30**(1): p. 79-82.

55. Hong, Y., W.Q. Meeker, and J.D. McCalley, *Prediction of remaining life of power transformers based on left truncated and right censored lifetime data.* The Annals of Applied Statistics, 2009. **3**(2): p. 857-879.

56. James, G., et al., *An introduction to statistical learning*. Vol. 112. 2013: Springer.

57. Gumpertz, M.L., J.M. Graham, and J.B. Ristaino, *Autologistic model of spatial pattern of Phytophthora epidemic in bell pepper: effects of soil variables on disease presence.* Journal of Agricultural, Biological, and Environmental Statistics, 1997: p. 131-156.

58. Burnham, K.P. and D.R. Anderson, *Model selection and multimodel inference: a practical information-theoretic approach*. 2003: Springer Science & Business Media.

59. Blitzer, J., R. McDonald, and F. Pereira. *Domain adaptation with structural correspondence learning*. in *Proceedings of the 2006 conference on empirical methods in natural language processing*. 2006. Association for Computational Linguistics.

60. Daumé III, H., *Frustratingly easy domain adaptation.* arXiv preprint arXiv:0907.1815, 2009.

61. Pan, S.J., J.T. Kwok, and Q. Yang. *Transfer Learning via Dimensionality Reduction*. in *AAAI*. 2008.

62. Zeiler, M.D. and R. Fergus. *Visualizing and understanding convolutional networks*. in *European conference on computer vision*. 2014. Springer.

63. Donahue, J., et al. *Decaf: A deep convolutional activation feature for generic visual recognition*. in *International conference on machine learning*. 2014.

64. Liu, X., et al. *Representation Learning Using Multi-Task Deep Neural Networks for Semantic Classification and Information Retrieval*. in *HLT-NAACL*. 2015.

65. Oquab, M., et al. *Learning and transferring mid-level image representations using convolutional neural networks*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.

66. Chu, B., et al. *Best practices for fine-tuning visual classifiers to new domains*. in *Computer Vision–ECCV 2016 Workshops*. 2016. Springer.

67. Chen, H.-Y. and J.-T. Chien. *Deep semi-supervised learning for domain adaptation*. in *Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on*. 2015. IEEE.

68. Glorot, X., A. Bordes, and Y. Bengio. *Domain adaptation for large-scale sentiment classification: A deep learning approach*. in *Proceedings of the 28th international conference on machine learning (ICML-11)*. 2011.

69. Baldi, P. *Autoencoders, unsupervised learning, and deep architectures*. in *Proceedings of ICML workshop on unsupervised and transfer learning*. 2012.

70. Aminisharifabad, M., Q. Yang, and X. Wu, *Deep Learning-Based Reliability Method for Complex Survival Data.* IEEE Transactions on Reliability, 2019(Submitted).

71. Nesterov, Y. *A method of solving a convex programming problem with convergence rate O (1/k2)*. in *Soviet Mathematics Doklady*. 1983.

72. Yu, Q. and J. Dong, *Generation of pseudo random numbers and estimation under Cox models with time-dependent covariates.* Journal of Statistical Computation and Simulation, 2016. **86**(14): p. 2727-2739.

73. Breslow, N., *Covariance analysis of censored survival data.* Biometrics, 1974: p. 89-99.

ABSTRACT

DEEP LEARNING BASED RELIABILITY MODELS FOR HIGH DIMENSIONAL DATA

by

MOHAMMAD AMINISHARIFABAD

August 2019

Advisor: Dr. Qingyu Yang

Major: Industrial Engineering

Degree: Doctor of Philosophy

The reliability estimation of products has crucial applications in various industries, particularly in current competitive markets, as it has high economic impacts. Hence, reliability analysis and failure prediction are receiving increasing attention. Reliability models based on lifetime data have been developed for different modern applications. These models are able to predict failure by incorporating the influence of covariates on time-to-failure. The covariates are factors that affect the subjects' lifetime.

Modern technologies generate covariates which can be utilized to improve failure time prediction. However, there are several challenges to incorporate the covariates into reliability models. First, the covariates generally are high dimensional and topologically complex. Second, the existing reliability models are not efficient in modeling the effect on the complex covariates on failure time. Third, failure time information may not be available for all covariates, as collecting such information is a costly and time-consuming process.

To overcome the first challenge, we propose a statistical approach to model the complex data. The proposed model generalizes penalized logistic regression to capture the spatial properties of the data. An efficient parameter estimation method is developed

to make the model practical in case of large sample sizes. To tackle the second challenge, a deep learning-based reliability model is proposed. The model can capture the complex effect of the data on failure time. A novel loss function based on the partial likelihood function is developed to train the deep learning model. Furthermore, to overcome the third difficulty, we proposed a transfer learning-based reliability model to estimate failure time based on the failure time of similar covariates. The proposed model is based on a two-level autoencoder to minimize the distribution distance of covariates. A new parameter estimation method is developed to estimate the parameter of the proposed two-level autoencoder model.

Various simulation studies are conducted to demonstrate the proposed models. The results show that the proposed models outperformed the traditional statistical and reliability models. Moreover, physical experiments on advanced high strength steel are designed to demonstrate the proposed model. As microstructure images of the steels affect the failure time of the steel, the images are considered as covariates. The results show that the proposed models predict the failure time and hazard function of the materials more accurately than existing reliability models.

AUTOBIOGRAPHICAL STATEMENT

Mohammad Aminisharifabad is a Ph.D. student of Industrial and Systems Engineering at Wayne State University, Detroit, Michigan. He received his Master's degree in Computational Molecular Biology/ Bioinformatics at the Saarland University in 2012. He has a bachelors' degree in Computer Engineering-Software from Yazd University, Yazd, Iran. His major research interests include Data Analytics, Machine Learning, Deep Learning and Reliability. His papers have been published in top conferences and journals such as Journal of Quality Technology (JQT) and received several awards.