SOME STUDIES ON RELIABILITY ANALYSIS OF COMPLEX CYBER-PHYSICAL SYSTEMS

A Dissertation
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Alex Vicente Davila Frias

In Partial Fulfillment of the Requirements
for the Degree of
DOCTOR OF PHILOSOPHY

Major Department:
Industrial & Manufacturing Engineering

July 2021

Fargo, North Dakota

North Dakota State University
Graduate School

Title

SOME STUDIES ON RELIABILITY ANALYSIS OF COMPLEX CYBER-PHYSICAL SYSTEMS

By

Alex Vicente Davila Frias

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

DOCTOR OF PHILOSOPHY

SUPERVISORY COMMITTEE:

Dr. Om Prakash Yadav
Chair

Dr. Nita Yodo

Dr. Val Marinov

Dr. Yildirim Suzen

Dr. Beena Ajmera

Approved:

July 6, 2021	Dr. David Grewell
Date	Department Chair

ABSTRACT

Cyber-physical systems (CPSs), a term coined in 2006 refers to the integration of computation with physical processes. Particularly, modern critical infrastructures are examples of CPSs, like smart electric power grids, intelligent water distribution networks, and intelligent transportation systems. CPSs provide critical services that have great impact in nation's economy, security, and health. Therefore, reliability is a primary metric. Nevertheless, the study of complex CPSs reliability demands understanding the joint dynamics of physical processes, hardware, software, and networks. In the present research, a series of studies is proposed to contribute to the challenging reliability analysis of CPSs by considering the reliability of physical components, hardware/software interactions, and overall reliability of CPSs modeled as networks. First, emerging technologies such as flexible electronics combined with data analytics and artificial intelligence, are now part of modern CPSs. In the present work, accelerated degradation testing (ADT) design and data analysis is considered for flexible hybrid electronic (FHE) devices, which can be part of the physical components or sensors of a CPS. Second, an important aspect of CPS is the interaction between hardware and software. Most of the existing work assume independency between hardware and software. In this work, a probabilistic approach is proposed to model such interactions using a Markov model and Monte Carlo simulation. Third, networks have been widely used to model CPSs reliability because they both have interconnected components. Estimating the network reliability by using traditional artificial neural networks (ANNs) has emerged as a promissory alternative to classical exact NP-hard algorithms; however, modern machine learning techniques have not been fully studied as reliability estimators for networks. This dissertation proposes the use of advanced deep learning (DL) techniques such as convolutional neural networks (CNNs) and deep neural networks

(DNNs) for all-terminal network reliability estimation problem. DL techniques provide higher accuracy in reliability prediction as well as the possibility to dispense with computationally expensive inputs such the reliability upper bound. In addition, most of the previous works assume binary states for the components of networks, whereas the present work incorporates a Bayesian method to consider degradation for network reliability estimation and updating of parameters.

ACKNOWLEDGMENTS

I wish to express my special thanks of gratitude and admiration to my advisor Dr. Om Prakash Yadav, who gave me the golden opportunity to work, with funding support, under his supervision on this exciting academic journey. I feel greatly thankful for his motivation, patience, guidance, and intellectual support during the whole learning and research process. His inspirational influence has significantly contributed to both my professional and personal development.

Besides my advisor, I would also like to thank to the rest of my committee members for their encouragement and insightful comments: Dr. Nita Yodo, Dr. Val Marinov, Dr. Yildirim Suzen, and Dr. Beena Ajmera. My sincere thanks to Dr. Marinov for the opportunity to work on a research project with his company.

Last but not least, I would like to thank all my family members and friends. This dream would not have been possible without their support and encouragement. Special thanks to my beloved wife Monica, my daughter Carolina, and my son Alejandro, for their love, patience, and understanding.

DEDICATION

This dissertation is dedicated to my mother and father, Adela and Vicente.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACA	Anisotropic conductive adhesive
ACP	Anisotropic conductive paste
ADT	Accelerated degradation testing
ALT	Accelerated life testing
ANN	Artificial neural network
BM	Bayesian method
CNN	Convolutional neural network
CPS	Cyber-physical system
DL	Deep learning
DNN	Deep neural network
FHE	Flexible hybrid electronics
GAMS	General algebraic modeling system
GEM	Graph embedding method
HW	Hardware
LCP	Liquid crystal polymer
LED	Light emitting diode
MC	Monte Carlo
MLE	Maximum likelihood estimate
NHPP	Non-homogeneous Poisson process
PET	Polyethylene terephthalate
PHM	Prognostics health management
RBD	Reliability block diagram
RFID	Radio frequency identification
RMSE	Root mean square error

RUL	Remaining useful life
S-L	Stress-life
SW	Software
Tg	Glass transition temperature

LIST OF SYMBOLS

$\mathrm{F}(\cdot)$	Cumulative distribution function
$\mathrm{f}(\cdot)$	Probability density function
$\mathrm{Pr}(\cdot)$	Probability function
$\mathrm{Pr}(\cdot \mid \cdot)$	Conditional probability
$\Phi_{\mathrm{nor}}(\cdot)$	Standardized normal CDF
D_f	Threshold level
D_{ijk}	Actual degradation y_{ijk} for sample i, at time t_j under the multi-stress level k
$Q_k(t)$	Probability of Markov process being at the k^{th} state at time t
d_i	Node degree of node n_i
p_L	Links reliability
p_N	Nodes reliability
y_{ijk}	Observed sample degradation y_{ijk} for sample i, at time t_j under the multi-stress level k
β_i	i^{th} degradation model parameter
$\varphi_{\mathrm{nor}}(\cdot)$	Standardized normal PDF
$C(n_i)$	Clustering coefficient of node n_i
$R(\cdot)$	Reliability function
$diag(\cdot)$	Diagonal matrix of a vector
$\mathbb{E}(\cdot)$	Expected value
$\mathbf{\Sigma}$	Covariance matrix
μ	Mean
σ	Standard deviation
ϵ	Residual deviation

$\boldsymbol{\mu}$ ....Mean vector

CHAPTER 1. INTRODUCTION

The term cyber-physical systems (CPSs) was coined in 2006 by Helen Gill at the National Science Foundation (U.S.) for the integration of computation with physical processes [1]. Applications domains include agriculture, aeronautics, building design, civil infrastructure, energy, environmental quality, healthcare and personalized medicine, manufacturing, and transportation [2]. Particularly, modern critical infrastructures are examples of CPSs, like smart electric power grids, intelligent water distribution networks, and intelligent transportation systems. [3]. They provide critical services that have great impact in nation's economy, security, and health.

Reliability is an important feature of CPS [4]. Moreover, it is critical because the system failures have enormous impact on business, environment and society [5]. For example, in the Northeast US blackout in 2003 more than 50 million people were affected [6, 7] and the estimated cost was about $ 6 billion [7]. Southwest blackout in 2011 and 2013 Central California blackout left without electricity to 7 million and 145 million people, respectively [3]. Moreover, due to interdependencies, other CPS like water distribution and transportation fail as a result of failures in power system CPS [8]. In addition to random failures, the interdependency with cyber components also expose CPS to cyber-attacks [9]. Due to automation mechanisms, cyber-attacks may reach to a large number of critical components [10]. For instance, outages in electric power systems may propagate between the coupled networks, increasing the risk of system wide cascading failures [11]. A blackout in 2015 was considered the largest one caused by cyberattacks for the first time in history [9]. Similarly, in 2016 a large number of computers were run offline in Israel's power supply system [12, 13]. Therefore, proper working of CPS is vital, and their reliability is a primary metric.

1.1. Overview

As a result of CPS complexity, reliability analysis of such systems is a cumbersome effort. The complexity of CPS is reflected on the different perspectives from which their reliability can be analyzed. Therefore, CPS reliability analysis problem can be addressed by understanding the reliability at components level and further integrating the concepts of components reliability into system reliability models. Since CPS integrate physical and cyber components, appropriate methods are developed for the reliability estimation of physical components considering ADT, and cyber components considering hardware/software interactions. In addition, reliability at network level is proposed for overall system reliability analysis, considering that CPS have interconnected components that may degrade. Therefore, in the present research, a series of studies is proposed to contribute to reliability estimation of CPS considering physical components, hardware/software interactions, and overall network reliability.

First, accelerated degradation testing (ADT) design and data analysis is considered to develop reliability estimation methods at components level. Flexible hybrid electronic (FHE) devices ADT data are used to demonstrate the developed methods, because they can be part of the physical components or sensors of a CPS. Nevertheless, the methods can be used with different kind of technology as well, with appropriate considerations of materials in test design. Our methods build the relations to estimate components reliability as a function of time and a developed equivalent stress relation. Our work is based on ADT, a developed multi-stress (S-L) relation, and probabilistic damage accumulation. As a result, we provide a framework for reliability estimation under realistic scenarios of multi-stress variables and multi-stress levels.

Second, an important aspect of CPS are the cyber components, which exhibit interaction between hardware and software. Most of the existing work assume independency between hardware and software or constant parameters to model their interactions. In this work, a probabilistic approach is proposed to estimate cyber components reliability, considering the interactions. Our work is based on NHPPs (non-homogeneous Poisson Processes), Markov model and Monte Carlo simulation. The realistic assumption of random parameters allows the estimation of confidence intervals on top of point reliability estimates as function of time.

Third, networks have been widely used to model CPSs reliability at system level because networks can capture the interconnection of CPS components. Estimating the network reliability by using traditional artificial neural networks (ANNs) has emerged as a promissory alternative to classical exact NP-hard algorithms; however, modern machine learning techniques have not been fully studied as reliability estimators for networks. This dissertation develops methods based on advanced deep learning (DL) techniques such as convolutional neural networks (CNNs) and deep neural networks (DNNs) for all-terminal network reliability estimation problem. DL techniques demonstrate higher accuracy in reliability prediction as well as the possibility to dispense with computationally expensive inputs such the reliability upper bound. In addition, most of the previous works on network reliability assume binary states for the links and perfect nodes. On the other hand, the present work proposes a framework based on DNN, MC, and Bayesian methods, to consider degradation of both links and nodes for network reliability estimation. Moreover, our proposed framework allows updating the parameters by incorporating new degradation data as they become available. Initial and updated point estimations of reliability, as function of time, are provided along with credible intervals.

In summary, the methods developed, and results obtained through this work suggest that due to the complexity of CPS reliability analysis, this problem can be addressed by understanding the reliability at components level, but also integrating the concepts of components reliability into system reliability models. For instance, a major contribution of our last framework is to integrate the concepts of component reliability and network reliability, by modeling nodes and links as components that degrade. This integration provides the methods for modeling and estimating the reliability at both component level and system level, by using degradation data that can be provided by modern sensor technology. Therefore, to address some of the challenges of CPS reliability estimation problem, this dissertation aims at the following research objectives:

- Objective 1: reliability assessment of physical components of a CPS considering ADT.
- Objective 2: reliability assessment of CPS considering hardware-software interactions.
- Objective 3: framework for network reliability assessment and updating of parameters considering degradation of links and nodes.

1.2. Objective 1: reliability assessment of physical components of a CPS considering ADT

Physical components of different CPSs can belong to a variety of classes. For instance, in electrical networks, generation, transmission, and distribution elements should be considered. In another example, a telecommunication network may consider switches and optical fiber as components. Although the components may differ among different CPSs, or even within the same CPS, a generic model is proposed for reliability assessment of components. Traditional ALT (accelerated life testing) may not be suitable for highly reliable modern components, which may hardly fail even during accelerated stress conditions. On the other hand, accelerated degradation testing (ADT) offers a more effective approach to estimate failure-time distributions even if there are no failures occurred during testing [14]. Emerging technologies such as flexible electronics combined with data analytics and artificial intelligence, are now entering the market today as part of modern CPSs [15]. In the present work, the ADT design, data analysis and reliability estimation under normal conditions, is proposed for flexible hybrid electronic (FHE) devices (Objective 1). FHE devices can be part of the physical components or sensors of a CPS. For instance, the flexible Radio Frequency Identification (RFID) tags have been used as sensors in CPSs [16, 17]. Moreover, RFID systems are often considered a subset of CPSs [18-21]. Although FHE devices will be used to demonstrate the developed ADT methods, most of them, with appropriate material specific considerations, could be applied to rigid electronics or other kind of components as well.

1.3. Objective 2: reliability assessment of CPS considering hardware-software interactions

One important aspect of CPS is the interaction between physical components, which degrade, with cyber components highly dependent on software that does not degrade but can fail. Although some studies have presented hardware-software reliability models, most of them assume independency between hardware and software. Some works aimed to capture interactions between hardware and software. Nonetheless, a common approach has been assuming independent series blocks, i.e., hardware and software, for reliability assessment [22-25]. This research work proposes to develop a reliability model for CPS that captures the changing interactions between hardware and software based on a probabilistic approach (Objective 2). Different from previous research, a probabilistic behavior will be assumed rather than a fixed ratio for software/hardware failures, which has been a common assumption in previous works [22-25], and does not necessarily reflects the reality [26-28].

1.4. Objective 3: framework for network reliability assessment and updating of parameters considering degradation of links and nodes

While reliability block diagram (RBD) is a widely used technique for system reliability assessment, it might result inadequate for complex multicomponent systems [5]. Therefore, networks have been commonly used to model CPSs, because networks [29-31] can capture the connectivity and interactions of CPS components [3, 13, 32-34]. For instance, electric power systems present interactions between the physical components (e.g., generators, transmission lines, transformers) and the communication components [11, 13, 32, 35].

Networks are usually represented by graphical models, where a graph $G\ (N, L)$ denotes the graph G composed by the set N of nodes and the set L of links or edges [36-38]. Regardless, the number of nodes, links, or their interconnection, the network reliability has several definitions, most of them associated with connectivity [32].

Three popular measures are all-terminal, two-terminal, and *k*-terminal [39]. All-terminal reliability is the probability that every node can communicate with every other node in the network. The two-terminal reliability problem requires that a pair of specified nodes, e.g., source and terminal, be able to communicate with one another. K-terminal reliability requires that a specified set of *k* target nodes be able to communicate with one another. Even though the two-terminal reliability problem is simpler than the all-terminal reliability one [40] and the *k*-terminal reliability is indeed a subset of all-terminal reliability with a space set restricted to *k* nodes only [29], advanced network reliability techniques have been focused on the all-terminal reliability [41-45]. Nonetheless, all-terminal exact reliability is an NP-hard problem [40, 42], which has led to the search of approximated but more efficient methods [46] such as Monte Carlo (MC) simulation [47-49] or bounds[40, 50, 51]. Since the exact reliability methods are computationally expensive, bounds algorithms appeared as an alternative to reduce the computational effort. Nonetheless, bounds methods still need the path sets and cut sets to be enumerated, requiring therefore exponential (in nodes) time [52]. Some improved algorithms to calculate the bounds are computationally intensive, e.g. $O(m^3)$ time [46], where m is the number of edges. MC methods provide a precise estimation of network reliability [47, 49]; however, they require numerous simulations to generate estimates. As a result, increased computational effort is required by MC methods, especially for highly reliable networks which rarely exhibit failures.

Estimating the all-terminal network reliability by using traditional artificial neural networks (ANNs) has emerged as a promissory alternative to classical exact NP-hard algorithms; however, modern machine learning techniques have not been fully studied as reliability estimators for networks. This dissertation proposes the use of advanced deep learning techniques such as convolutional neural networks (CNNs) and deep neural networks (DNNs) for all-terminal network reliability estimation problem. Features such as an adaptive learning rate, rate annealing, momentum training, dropout, and regularization, are believed to contribute to better generalization compared to traditional ANNs. DL techniques provide higher accuracy in reliability prediction as well as the possibility to dispense with computationally expensive inputs such the reliability upper bound.

Another limitation of existing network reliability estimation method is the assumption of discrete functional states. In previous research of CPS modeled as networks, the majority of the studies have assumed binary states, i.e. functional or failed, for nodes [3, 32, 53, 54] or edges (links) [40, 55]. As an alternative to solve the network reliability problem, Markov models have been proposed for reliability evaluation of CPS [4, 35, 56-58]. One of the main drawbacks of Markov models is that few assumed discrete states do not necessarily mimic continuous degradation state of real systems due to components degradation, whereas increasing the number of states makes the computation more complicated. Therefore, there is a need to explore system reliability modeling and assessment considering continuous degradation of the components. Degradation can be modeled by either general path [59] or stochastic processes [60], e.g., Wiener [61, 62], Gamma [63-65] and Inverse Gaussian (IG) [66-68].

In addition, if the network reliability is regarded as dependent not only on its topology [69] but also on raw degradation data from nodes and edges, a vast amount and variety of data may be generated. Therefore, appropriate methods need to be developed to capture the network topology and topological attributes in a manner that they can be fed deep learning systems along with degradation data.

A holistic approach for reliability assessment is proposed to integrate degradation data from nodes and edges with network topology information to estimate the reliability of networks. The amount and variety of data leads to high accuracy in the reliability estimation. Nevertheless, the increment in the size and complexity of data demands the use of modern Bayesian Methods (BM) and machine learning techniques such as CNN or DNN.

Objective 3 of this dissertation aims to develop a framework to estimate network reliability considering degradation data of both nodes and edges. To address this problem, an integration of BM, MC and DL methods in a framework is proposed.

1.5. Dissertation organization

CHAPTER 2 presents a summary of the relevant literature.

CHAPTER 3 provides a generic model for reliability prediction of physical components considering multi-stress variables and multi-stress levels. The model is based on ADT design and data analysis. FHE devices are used to demonstrate the developed ADT methods. This chapter is based on three papers and supports objective 1. A generic model for reliability prediction considering more than one stress variable and multi-stress levels was proposed in our paper recently published in *ASME Journal of Electronic Packaging* [70]. This work was preceded by an extensive review in methods for reliability testing of FHE, published in the journal *IEEE Transactions on Components, Packaging and Manufacturing Technology* [71]. Another input for the model was provided by ADT design and data analysis, published in our paper in the *International Technical Conference and Exhibition on Packaging and Integration of Electronic and Photonic Microsystems* [72].

CHAPTER 4 is focused on a probabilistic approach for reliability assessment of CPS considering hardware-software interactions. This chapter is based on two papers and supports objective 2. Our paper about CPS reliability considering hardware and software components, was presented at the *Reliability and Maintainability Symposium (RAMS) 2019* [73] and is published in the *Proceedings of the Annual Reliability and Maintainability Symposium 2019*. In addition, our paper about system reliability considering probabilistic hardware/software interactions was submitted to *IEEE Transactions on Systems, Man, and Cybernetics: Systems* [74].

CHAPTER 5 proposes two all-terminal network reliability estimation approaches. One approach is based on CNN and the other approach integrates DNN with GEM. This chapter is based on two of our papers and describes a preliminary step in the development of an overall framework for all-terminal network reliability assessment proposed in objective 3. Our paper about network reliability estimation using CNNs was published in the *Journal of Risk and*

Reliability [75]. Also, our paper *'Deep Neural Networks (DNNs) For All-Terminal Network Reliability Estimation'* was presented at the *Reliability and Maintainability Symposium (RAMS May 2021)* and accepted for publication in *Proceedings of the Annual Reliability and Maintainability Symposium 2021* [76]. This work was selected as the winner of the 2021 Stan Oftshun Award for the best student paper by the Society of Reliability Engineers (SRE).

CHAPTER 6 presents two studies on all-terminal network reliability that relax the common assumption of constant (or even perfect) reliability of links and nodes. This chapter is based on two of our papers and supports objective 3. Our paper, presented in the *IISE 2021 Annual Conference* [77], considers the relaxation of the assumption of perfect nodes on the network reliability estimation. The approach presented in such paper integrates MC and DNN, which allows fast network reliability estimation. Finally, we present a framework for network reliability estimation of parameters considering degradation of links and nodes. The proposed approach integrates BM, MC simulation, and DNN in a framework to estimate the reliability of networks as a function of time. Not only point estimates are provided but also credible intervals. Moreover, the proposed framework provides a mechanism for updating the model parameters as new information becomes available. This work was submitted to the journal *Reliability Engineering and System Safety* [78].

Finally, CHAPTER 7 summarizes the whole work and provides the future research direction.

1.6. Statement of authorship

Most of the content of the chapters presented in this dissertation is based on the following journal and conference papers, which are published, accepted, or submitted for publication:

1.6.1. Peer-reviewed journal papers

1. Davila-Frias, S. Limon, V. Marinov, and O. P. Yadav, "Reliability Evaluation of Flexible Hybrid Electronics Systems Considering Degradation Behavior Under Multistress Operating Conditions,". Published in *Journal of Electronic Packaging,* vol. 143, no. 2, 2020, doi: 10.1115/1.4048035. [Used in CHAPTER 3].
2. A. Davila-Frias, O. P. Yadav, and V. Marinov, "A Review of Methods for the Reliability Testing of Flexible Hybrid Electronics,". Published in *IEEE Transactions on Components, Packaging and Manufacturing Technology,* 2020. [Used in CHAPTER 3].
3. A. Davila-Frias, N. Yodo, and O. Yadav, "Probabilistic modeling of hardware and software interactions for reliability prediction of embedded systems," 2019. In *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. Submitted. [Used in CHAPTER 4].
4. A. Davila-Frias and O. P. Yadav, "All-terminal network reliability estimation using convolutional neural networks,". Published in *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability,* p. 1748006X20969465, 2020. [Used in CHAPTER 5].
5. A. Davila-Frias, N. Yodo, T. Le, and O. P. Yadav, " A Deep Neural Network and Bayesian Method based Framework for All-Terminal Network Reliability Estimation Considering Degradation" 2021. In *Reliability Engineering and System Safety*. Submitted. [Used in CHAPTER 6].

1.6.2. Peer-reviewed conference papers

1. A. Davila-Frias, V. Marinov, O. P. Yadav, and Y. Atanasov, "Design of Accelerated Degradation Test Method and Failure Analysis of Flexible Hybrid Electronic Devices,". Published in *International Electronic Packaging Technical Conference and Exhibition*, 2020, vol. 84041: American Society of Mechanical Engineers, p. V001T02A001. [Used in CHAPTER 3].
2. A. Davila Frias, N. Yodo, and O. P. Yadav, "Mixed-Degradation Profiles Assessment of Critical Components in Cyber-Physical Systems,". Published in *2019 Annual Reliability and Maintainability Symposium (RAMS)*, 28-31 Jan. 2019, pp. 1-6. [Used in CHAPTER 4].
3. A. Davila-Frias, S. Salem, and O. P. Yadav, "Deep Neural Networks (DNNs) For All-Terminal Network Reliability Estimation," 2021. In *2021 Annual Reliability and Maintainability Symposium (RAMS)*. Presented. **2021 Stan Oftshun best student paper award by the Society of Reliability Engineers (SRE).** [Used in CHAPTER 5].
4. A. Davila-Frias, O. P. Yadav, S. Salem, and B. Nepal, "All-Terminal Network Reliability Estimation with Monte Carlo and Deep Neural Networks," 2021. In *IISE 2021 Annual Conference*. Presented. [Used in CHAPTER 6].

CHAPTER 2. LITERATURE REVIEW

Reliability definition (section 2.1) and CPS modeling (2.2) overview are presented as a prelude to the literature related to reliability of CPS (2.3).

2.1. Reliability definition

Reliability is the probability (or the ability) of a system, sub-system or component "to perform a required function, under given environmental and operational conditions and for a stated period of time" [79-81]. Nevertheless, due to the complexity of CPS, reliability analysis has been addressed from different perspectives [4]. As a result, complementary or specific reliability metrics have been employed depending on CPS modeling approaches or particular functionality. For instance, networks have been widely used to model CPS, and the most popular measures of network reliability are associated with network connectivity [32], thus the reliability can be defined as the probability that surviving nodes and edges span an operating subnetwork [29]. One interpretation is related to how many of the n nodes can communicate with each other (e.g. two-terminal, k-terminal or all-terminal problems) [39, 40], assuming that a number of the links can fail. However, in a more general model, the nodes can fail as well and the node reliability is the probability of being operational over a time period [32].

2.2. Modeling of CPS

Reliability assessment approaches depend on the assumed models for CPS, which can be broadly classified in four categories: graphical, functional, probabilistic, and simulation based.

2.2.1. Graphical models

Graphical models allow to visualize the interdependencies of the components. Such models are representations of physical objects in terms of nodes and edges, where nodes characterize components and junctions of the system, and edges represent the connections. For example, busbars in power systems or switches in telecommunication systems are modeled by nodes, whereas edges characterize power lines in power systems and optical fibers in telecommunication systems. These models are commonly based on graph theory (GT), where a graph *G (N, L)* denotes the graph *G* composed by the set *N* of nodes and the set *L* of links [36-38]. Furthermore, to model the complexity of some CPS, several layers, e.g., cyber, physical, and interface, could be considered and result in complex interconnected networks [54, 82-86]. Zio et al [87] considered both electrical (transmission lines) and networks components to analyze power grids from four different perspectives, namely, topological, reliability, electrical, and electrical-reliability. Finite state machine models [1, 88], and petri net models [89, 90] have been used as well, being computational exponential time increasing with system size, a major drawback.

2.2.2. Functional models

They attempt to capture the physical system dynamics [91, 92] and the response of the systems exposed to strains [93]. Load flow for power systems, pressure for water systems, or connectivity-based models, are examples of these kind of models. [93]. Some of the models consider the sampling in the control loops [94, 95], since the signals should be discretized to flow through the communication networks.

In general, functional models capture the physical behavior in a more realistic manner than the graphical (topological) models. However, topological models are computationally very fast [96-98]. Some functional models sacrifice accuracy to reduce computational time. For example, approximated DC power flow is most commonly used than AC model because the last one is usually time consuming and do not lead to convergence into global optimal solutions [38, 99, 100].

2.2.3. Probabilistic models

CPS may be affected by probabilistic events, which can be modeled by uncertainty models. Han et al [101] studied the impacts of cyber system probabilistic failures on the availability of physical components. Overall CPS reliability can be summarized in probability tables based on reliability information of components [102, 103]. State transition probability of components can be modeled by state transition diagrams [104, 105]. Furthermore, Bayesian network (probabilistic) approaches have been used for power system reliability assessment [106], reliability analysis of complex systems [107], and recently, dynamic Bayesian network combined with hidden Markov model were used to estimate and predict functional reliability of a complex chemical process [5]. On top of probabilistic failures of CPS components, CPS depend on rational decision-making participants, e.g. system operators and hackers, situation that can be studied by game-theoretic models [108, 109].

2.2.4. Simulation models

Numerical simulation models aim to preserve the accuracy of complex CPS considering the discrete characteristics of cyber systems integrated with the continuous nature of physical systems. Such models have been developed on different software packages. For instance, for power physical systems: Matlab [110, 111], OpenDSS [112], DIgSILENT [113]. On the other hand, languages like Java [114] and Visual Studio [115] can be employed to create cyber systems simulation models. Simulation is often a suitable approach since it is impractical to test in real CPS, e.g., power grids [11]. In addition, there are some reports on previous blackouts, but they do not offer details about hardware and software components failures. Moreover, due to the high reliability of smart grids, many potential scenarios have never occurred in the past [3].

2.3. CPS reliability, related research works

In CPS, the physical processes are under the control and monitoring of embedded computers and networks, usually through feedback loops where physical processes affect computations and vice versa [1]. A simple composition model considers that a CPS comprises a physical component or hardware, cyber component or software, and communication among them. The system CPS reliability can be estimated as an integration of its components' reliability [116]. In an effort to assess the reliability of a CPS, it is imperative to understand the available models to estimate the reliability of their components. As a starting point, a CPS can be decomposed in physical components (or subsystems) and software components (or subsystems).

Regarding physical components, the reliability metrics could be derived through ALT or ADT. With the advancement in products with high reliability, failures will hardly occur prematurely even under high-stress conditions, which make ALT tests impractical in many cases. On the other hand, ADT offers an approach to estimate failure-time distributions even if there are no failures occurred during testing [14]. ADT test design, data analysis and reliability estimation under normal conditions, considering multi-stress levels and multi stress variables is proposed for the present research (Objective 1).

Reliability assessment based on multiple components degradation has been already studied [117, 118]. The cyber components, which are associated with the software perspective, require different approach to assess their reliability. Since there is no degradation or wear out in software, predictive approaches based on software testing failure data have been developed to estimate software reliability [119-122].

A unified hardware-software reliability model was introduced by Welke et al. [123]. The model incorporates the time-varying software failure intensity of the Goel-Okumoto/NHPP (non-homogeneous Poisson Process) model [119] into a Markov hardware reliability model. Similarly, Teng et al. [22] proposed a reliability model which combines Weibull distribution for hardware time-to-failure and a Gokel-Okumoto/ NHPP model with imperfect debugging for software failures. In the same way, Roy et al. [23] applied a unified hardware–software reliability model, which combines Weibull modelling for hardware failures and NHPP for software failures, to a smart grid physical infrastructure. Feng et al. [24] proposed a unified hardware-software model based on hardware failures, software failures, and assumed that 75% of hardware failures are indeed hardware/software interaction failures whereas 35% of the software failures are hardware-related. Koc [25] studied the reliability of a CPS assuming constant reliability data of both physical and cyber components. Previously proposed combined models are focused on hardware-software computer systems and commonly depend on failure data for hardware components, not on degradation. Nannapaneni [53] presented a reliability analysis framework for a smart parking CPS considering the timing constraints in addition to the usual operational requirements. The uncertainty in time of communication network was modeled by a probability density function and Monte Carlo sampling. Constant failure rates were assumed for the hardware components, and software was assumed to always work (rigorous testing) without failure.

Research work was conducted by the author of this dissertation [73] to estimate the overall reliability profile of a CPS based on the mixed-degradation profiles of its physical components and the reliability profile of its software component that is responsible in controlling the CPS. A CPS with several critical physical components was assumed to follow different degradation path functions (linear, exponential, polynomial, etc.) and shaping a physical subsystem. A major critical software component was assumed, whose reliability was assessed by utilizing one of the latest available NHPP software reliability models. For the CPS to operate without failure, it is required for both physical and software subsystems to function properly. That is to say, physical subsystem was assumed to be in series with a major independent cyber component, i.e., a reliability block diagram (RBD) approach was used.

Although some studies have presented hardware-software reliability models, most of them assume independency between hardware and software. Some works aimed to capture interactions between hardware and software. Nonetheless, a common approach has been to assume independent series blocks, i.e., hardware, software, hardware/software, and software/hardware, for reliability assessment. Moreover, constant proportion of failures has been commonly assumed; e.g., 35% of observed software failures have been assigned as hardware-related failures [22-25]. However, the classification of 35% of observed software failures as hardware-related comes from a previous result [124] and does not necessarily captures hardware-software interactions of current systems, where the ratio of hardware-related software failures is not necessarily a fixed value [26-28]. In addition to the already published work, this research project proposes to develop a reliability model for CPS that captures the changing interactions between hardware and software based on probabilistic models (Objective 2).

While RBD is the most widely used technique for system reliability assessment, it might result inadequate for complex multicomponent systems [5]. A major limitation is usually the lack of sufficient failure data for all system components [125]. Moreover, RBD approach allows only two functional states for system components, e.g. perfectly working or failed [5]. However, binary levels of failure does not necessarily mimic real-world situations [126, 127]. To capture CPS complexity, networks have been used as a promising modeling alternative.

A network can be defined as a set of items (nodes or vertices) connected by edges or links [128]. Usually, networks are modeled as mathematical graphs composed of nodes and links (directed or undirected) and used to represent physical problems such as computer networks, piping systems, or power supply systems. In such models, the links have associated parameters like flow (m^3/s), bandwidth (Mbps), or power (MW) and provide communication pathways to the nodes, which represent users or resources [40].

Although networks are not a new topic, they have been widely used to model CPS, thus some network reliability approaches are included in this review. Moskowitz [55] proposed a network model with perfect vertices and edges either operational or failed. He developed an exact algorithm for network reliability as a function of the reliability of two networks. Theologou and Carlier [129] continued this work by proposing a factoring algorithm and reductions for reliability calculation considering imperfect vertices. Wood [29] contributed to reduce computation to the network reliability problem, which is known to be NP-hard, with a reliability-preserving model based on reduction methods. Wood assumed perfect vertices and edges with independent operation probabilities. Taking into consideration the complexity of exact network reliability calculation, Aypub et al. [30] developed an algorithm based on Monte Carlo simulation and breadth-first search to get accurate estimates of reliability in feasible and practical time. Cancela et al. [31] proposed a polynomial-time algorithm for deleting irrelevant edges based on the evaluation of the source-terminal diameter constrained problem (source-terminal length should not be greater than D). They combined this procedure with Moskowitz's exact method to calculate the reliability and obtained an important computational gain. As a part of the machine learning techniques, artificial neural networks (ANNs) have emerged as a promissory tool to estimate network reliability. Indeed, ANNs have been claimed to be one of the most efficient methodologies developed so far for reliability estimation of networks [130]. ANNs have been usually trained with the network topology and edge reliability as inputs and with the target network reliability as desired output [42, 45, 131]. For example, Srivaree- ratana et al. [42] utilized an ANN to predict the all-terminal network reliability; with the network architecture, the link reliability, and the reliability upper bound as inputs, and the exact network reliability as the target. More recently, Altiparmak et al. [45] proposed an ANN model to predict the all-terminal network reliability, which takes the node degree and other connectivity metrics and the upper bound network reliability as inputs to predict the network reliability. Similarly, Dash et al. [131] proposed a method based on ANNs to maximize the reliability of fully connected networks subjected to some predefined total cost. In the present dissertation, the network topology and topological attributes will be considered for appropriate preprocessing and formatting for they to be used during deep learning training. Deep learning techniques such as CNN and DNN have not been fully explored for network reliability and they will be considered for the present study.

In addition, in previous research of CPS modeled as networks, the majority of the studies have assumed binary states, i.e. functional or failed, for nodes [3, 32, 53, 54] or links [40, 55]. As an alternative to overcome this problem, Markov models have been proposed for reliability evaluation of CPS.

Wu et al. [56] proposed Markov models for each component to assess the reliability of an integrated modular avionics system. Marashi et al. [57] modeled a smart grid with dependent components with a Markov imbedded system and computed the probability that the system is in one of the functional states. Sun et al. [58] proposed a phased-mission system model, that is Markov models for individual components and a binary decision diagram to analyze the reliability of a fuel management system of an aircraft. Li and Kang [4] developed a multi-index method considering individual reliability of components and performance reliability indices for service, cyber security, resilience & elasticity, and vulnerability. Recently, Gunduz and Jayaweera [35] proposed a model to assess reliability of cyber-physical integrated system operation with multiple photovoltaic (PV) system configurations by incorporating Markov-Chain transitions for PV system components. One of the main drawbacks of Markov models is that few assumed discrete states do not necessarily mimic continuous degradation state of real systems due to components degradation, whereas increasing the number of states makes the computation more complicated.

Therefore, there is a need to explore system reliability modeling and assessment considering continuous degradation of the nodes and the edges of a network. One of the objectives of this dissertation aims to address this problem (Objective 3) by developing a framework to provide reliability assessment and a mechanism for updating model parameters as more information becomes available. Physical components could be for example electrical or mechanical and therefore exhibit different degradation profiles which need to be incorporated into the network model to estimate the reliability of the CPS.

The objective 3 of this dissertation seeks addressing this challenge. Appropriate input preprocessing and formatting, training and deep learning system architectures need to be developed. The complexity of the generated degradation data is expected to produce vast and diverse data which will require advanced machine learning methods to deal with them. An integration of BM, MC, and DL techniques is proposed to estimate the reliability of a network based on degradation data. In addition, the proposed framework is able to incorporate new available information for updating model parameters by applying a Bayesian approach.

2.4. Degradation modeling

Since this dissertation contemplates to consider degradation of links and nodes of networks, a brief review on degradation is presented in the next paragraphs.

Although failure data have been traditionally used for decades to estimate reliability, modern highly reliable products rarely result in failures even under stress application. Therefore, to assess reliability quickly, approaches based on degradation data have attracted attention to researchers. Measuring the performance characteristics of a product and their degradation over time provides a large amount of useful and essential information for assessing product reliability [132]. Moreover, degradation (data) models have been recently used for reliability assessment, remaining useful life (RUL) prediction, maintenance planning, and prognostics health management (PHM) [133]. Modern sensor technology facilitates the collection of degradation (of performance characteristics or related variables) data in both in-house testing facilities and in the field as well as covariates data like temperature, humidity, voltage, etc., which provide information about the operational environment.

Degradation modeling, prognostics, and health management can be divided into two broad categories: physics-based and data-driven. [67, 134]. Data-driven models are becoming popular because in reality it is difficult to capture the exact physics-of-failure. A more detailed classification is shown in Figure 1.

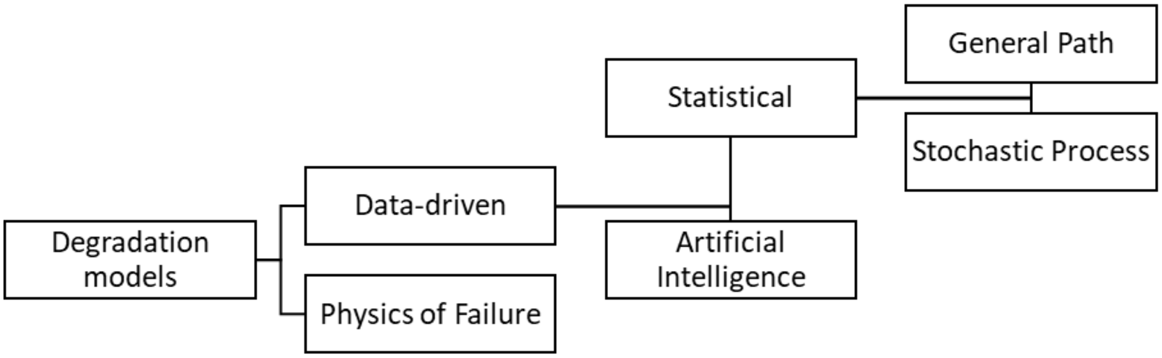


Figure 1. Degradation models

Considering statistical models, the degradation can be studied trough general path [59] or stochastic processes [60], e.g., Wiener [61, 62], Gamma [63-65] and Inverse Gaussian (IG) [66-68].

Degradation analysis can be extended to CPS components. Modern components of systems are equipped with sensors to monitor operation indicators as well as environmental conditions [135]. Locomotive engines, for example, have sensors of oil pressure, oil temperature, and engine coolant temperature, which generate time series data that are used to control the engine during normal operation and can be further used to warn of dangerous operating conditions [136]. Similarly, aircraft engines have sensors for operation temperature, oil temperature, debris, and vibration, which could be used for engine health monitoring [137]. Currently, even modern automobiles have data acquisition and communication systems like GM’s Onstar which can collect and upload operational and environmental indicators [135]. Power distribution transformers now can be outfitted with sensors to provide environmental and operational information in real time or with the required periodicity. For instance, dissolved gas analysis (DGA) can be applied to indicate possible transformer faults [138]. Wind turbine systems contain sensors to capture information on variables like relative movement, sway, and vibration, which can be used to perform system health monitoring (SHM) [139-141].

CHAPTER 3. RELIABILITY ASSESSMENT OF PHYSICAL COMPONENTS OF A CPS CONSIDERING DEGRADATION DATA[1]

Most of previous research on ADT is focused on single stress. A generic model for reliability prediction considering more than one stress variable and multi-stress levels is proposed in this chapter, based on ADT test design and data analysis. FHE devices are chosen for demonstration as FHE is an emerging technology that combined with data analytics and artificial intelligence, is now entering the market today as part of modern CPSs [15]. Although FHE devices will be used to demonstrate the developed ADT methods, most of them, with appropriate material limits considerations, could be applied to rigid electronics or other kind of components as well. Section 3.1 presents a review on FHE testing methods, section 3.2 discusses the ADT design, and section 3.3 proposes the model for reliability assessment considering ADT data.

[1] The present chapter is based on the following papers:

1. A. Davila-Frias, S. Limon, V. Marinov, and O. P. Yadav, "Reliability Evaluation of Flexible Hybrid Electronics Systems Considering Degradation Behavior Under Multistress Operating Conditions,". Published in Journal of Electronic Packaging, vol. 143, no. 2, 2020, doi: 10.1115/1.404803.
   Contribution of Alex Davila Frias: developing the mathematical models, analysis of the case study, discussion of the results, and drafting the paper.
   Contribution of Shah Limon, Val Marinov, and Om Yadav: verification of the results and proofreading the draft paper.

2. A. Davila-Frias, O. P. Yadav, and V. Marinov, "A Review of Methods for the Reliability Testing of Flexible Hybrid Electronics,". Published in IEEE Transactions on Components, Packaging and Manufacturing Technology, 2020.
   Contribution of Alex Davila Frias: conducting literature review, discussion, and drafting the paper.
   Contribution of Om Yadav and Val Marinov: proofreading the draft paper.

3. A. Davila-Frias, V. Marinov, O. P. Yadav, and Y. Atanasov, "Design of Accelerated Degradation Test Method and Failure Analysis of Flexible Hybrid Electronic Devices,". Published in International Electronic Packaging Technical Conference and Exhibition, 2020, vol. 84041: American Society of Mechanical Engineers, p. V001T02A001.
   Contribution of Alex Davila Frias developing test design, test setup, and mathematical models. Discussion of the results and drafting the paper.
   Contribution of Val Marinov, Om Yadav, and Yuriy Atanasov: verification of the results and proofreading the draft paper.

3.1. Review on FHE testing methods [71]

This section refers to an overview of testing methods applied to Flexible Hybrid Electronics (FHE). It describes general accelerated testing methods and how they are related to the current progress in FHE testing. Stressors like temperature cycling, temperature/humidity, bending, stretching and torsion have been commonly applied to FHE. The application of those stressors and the corresponding failure modes, mechanisms, and factors that impact reliability are summarized and discussed. Methods for testing the reliability of FHE devices such as RFID tags have been developed, but there are no industry-wide standards yet to address all aspects of this emerging technology.

Flexible Hybrid Electronics (FHE) describes a technology that combines flexible high-performance components (e.g., flexible silicon chips) with printed components (e.g., interconnects, sensors, or microfluidic channels) on non-traditional substrates (e.g., polymers and fabrics) that can flex, bend, stretch, and/or fold [142]. Section 3.1.1 focuses on traditional stressors for electronic testing that have been also applied to FHE, whereas section 3.1.2 highlights specific application FHE testing beyond traditional rigid electronics testing methods.

3.1.1. Traditional stressors

Even though FHE is a relatively new technology, it is still an electronic assembly. Therefore, it is reasonable to consider the main sources of stress for electronic products in general. Considering stress as the effect that usage and environmental loads place on a device and its materials, some of the typical stress sources are temperature, humidity, and vibration. Apart from these stressors there are also others, like altitude and corrosion, that are applicable to electronic testing [143]. Temperature is a key stress in most of ALT (Accelerated Life Testing) plans [144], not only for electronic products. Moreover, temperature, humidity and vibration are important sources of stress experienced in electronic equipment as displayed in Figure 2. They account for approximately 55%, 20%, and 19%, respectively, of the total frequency of occurrences of these stress sources [145]. These general percentages may of course vary, depending on the application and product location.

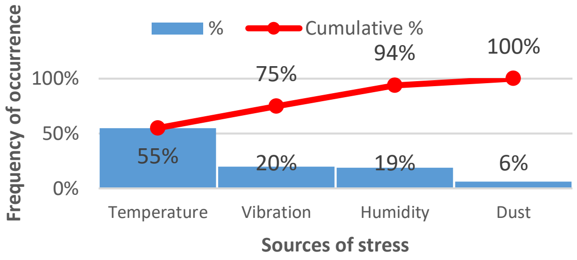


Figure 2. Sources of stress for electronic equipment as percentages of frequency of occurrences. Adapted from [145]

Vibration and mechanical shock tests have been applied to rigid circuit boards for they have limited bending and stretching capabilities [146]. However, due to the nature of FHE, bending , stretching [147-155], and/or twisting [156-158] are more appropriate factors and have been commonly evaluated, in addition to temperature and humidity.

Reported results show that the reliability, usually presented as cumulative distribution functions, depends on factors such as substrate thickness, chip thickness, characteristics of adhesive, type of chip, and stress loading.

Degradation in ACA (Anisotropic conductive adhesive) has been a common failure mechanism reported in several thermal cycling studies, and daisy-chain resistance is a common performance variable monitored. For instance, thermal cycling was applied to FHE (Flexible LCD with ultra-thin chip + PI + ACA) [159], and the failure mode was resistance increasing (daisy-chain) with failure mechanisms: expansion strain and warpage of chip, and degradation of electrical conductivity of conductive particles in ACA joints. In another study of FHE (Flexible

LCP (Liquid crystal polymer) substrate + ACA bonded flip chips) [160], the failure mode was also resistance increasing with failure mechanism: adhesive matrix deformed, the conductive particles lost contact and the joint failed. Similarly, different settings of thermal cycling [161-163] applied to RFID tags (PET + chip + ACP) led to an increase in the threshold power as a failure mode, with failure mechanisms of relaxation of ACA material and cracks in antenna wiring. The reliability of FHE devices, as we can expect, has been affected by aspects such as the thickness or glass transition temperature (Tg) of the components, which has been corroborated by several studies [159-163]. Another factor that, expectedly, impacts reliability is the stress loading [162-164].

Temperature combined with humidity has been applied for FHE testing [159, 165-169] as well. The failure mode was resistance increase (daisy-chain). Failure mechanisms found were expansion strain and warpage of chip, humidity absorption of substrate and adhesive matrix, delamination in ACA joint, and corrosion of the metallization.

Temperature/humidity have also been applied in RFID testing [170]. The failure mode was contact resistance increasing with a failure mechanism of resin moisture absorption followed by oxidation of aluminum pads. Another failure mode in RFID was threshold power increase [161, 164, 171]. The failure mechanisms were mainly adhesive matrix swelling that led to increased resistance in open joints and cracks in the wiring of the antenna near the RFIC (radio frequency chip).

Bending tests have been commonly applied to FHE. There are two broad categories of bending tests: static and dynamic. In static bending, the devices under test are wrapped around cylinders with various radii [172, 173]. In another approach of static tests, the specimen is gradually bent using a 3 or 4-point bending fixture until it fails [174-178]. Regarding dynamic bending, there are no standard test procedures. Manufacturers and researchers have developed their own testing methods subjecting the specimens to a number of bending cycles and radii [179-181]. Both static [159] and dynamic [179-181] tests have been applied to FHE. A common failure mode is daisy-chain or contact resistance increase associated with the failure mechanism peeling off the chip from the substrate. In the same way, both static [170, 182], and dynamic [182-185] bending tests have been applied to RFID as well.

3.1.2. Specific application FHE testing

Temperature cycling and temperature/humidity tests, traditionally used for regular rigid electronics, seem to be still applicable to FHE, whereas additional bending or twisting tests are required for FHE products due to their nature. Another similarity between regular electronics and FHE testing is the use of daisy-chain resistance to monitor the performance. In rigid electronics industry, daisy-chain resistance provides a simple and cost-effective testing mechanism to monitor deterioration. Daisy-chain resistance for FHE testing has been a popular performance indicator as well [159, 165-169, 181] as discussed in section 3.1.1. Nevertheless, traditional criteria such as the usual 20% resistance increase from IPC-9701A [186] may not be appropriate for FHE in some cases [152]. For instance, bending tests induce temporal strains, which may affect the electrical resistance. During the tests resistance may easily increase up to 20% [159] or even exceed 100% [152, 187]. However, after the stress removal, the original mechanical dimensions of electrical paths and consequently their related resistance values may be recovered. Nevertheless, if the resistance or any other electrical variable of a final product must exhibit a stable behavior, even under bending stress, a 20% variation could be still unacceptable. For example, a thermistor [158] sensor circuit may generate intolerable signal errors due to a bend/stretch force causing a resistance increase.

3.2. ADT test design and failure analysis of FHE devices [72]

In this section, we present the design and experimental setup of ADT for FHE considering two stress factors simultaneously. We use daisy-chain resistance as a measurable degradation characteristic to periodically monitor the degradation of FHE products under accelerated stress conditions. Two stress factors, temperature and humidity, are considered and ADT will be carried out considering four combinations of temperature and humidity simultaneously. Failure analysis will be performed on failed units to investigate the failure process and location of the failure. The ADT data will be used to fit in the appropriate mathematical degradation model representing the failure process.

Besides degradation modeling under stress conditions, an important goal of ADT is the estimation of product life under normal operating conditions. To this end, life-stress models are needed. Four stress levels combinations proposed would be enough for fitting life-stress models with up to four parameters, such as the generalized Eyring model. Eyring model has been a popular choice to capture the combined effect of temperature and non-thermal stress [188, 189]. It requires at least four combinations of stress levels to be estimated from the data [190] (see Table 1).

Table 1. Experimental design

Treatment combination $(\boldsymbol{k})$	Temperature [°C]	Rel. Humidity [%]	Duration [hours]
1	85	98%	150
2	85	85%	150
3	65	98%	150
4	65	85%	150

3.3. Reliability evaluation of FHE systems considering degradation behavior under multi-stress operating conditions [70]

Predicting the reliability is an important task of product life cycle analysis, especially during the product development stages. The uncertainty in the operating conditions and the presence of multi-stress factors makes this reliability prediction more difficult. In this section, a probabilistic reliability prediction framework is proposed using the linear damage accumulation and degradation modeling for multi-stress conditions. The multi stress-life (S-L) curve for corresponding multi-stress will be developed using the equivalent stress. The multi S-L model allows extrapolating the expected life product under given operating or test conditions as well as provides input to estimate the reliability as a function of time for both a single multi-stress factor level and a sequence of multi-stress-factor levels. The proposed methodology will be demonstrated with newly developed flexible hybrid electronics products.

The degradation path modeling is briefly outlined below:

Considering data from preliminary experiments, we propose to use a nonlinear exponential degradation model. Similar exponential models have been utilized to fit degradation data for contact resistance degradation in MOSFETS [191], and have proved to fit well cumulative degradation data [192]. Moreover, exponential degradation patterns have been observed in the degradation of electronic components [193]. Additionally, the moisture absorption of flexible substrates accelerated by temperature [194, 195] has shown an exponential degradation of adhesion to chips [196]. Daisy-chain resistance of FHE with ACA adhesive has shown nonlinear degradation paths [197]. Although in this particular case an exponential model is used to model the degradation process, as a part of the general framework, other degradation models could also be used depending on the observed degradation data.

Consider the actual degradation path of a particular unit over time is denoted by $D(t), t > 0$ Samples are observed at discrete points in time $t_1, t_2, \dots t_j, \dots$. The observed sample degradation y_{ijk} for sample i, at time t_j under the multi-stress level k in a general degradation path model is given as:

$$y_{ijk} = D_{ijk} + \epsilon_{ijk} \tag{1}$$

where $D_{ijk} = D(t_{ijk}, \beta_{1ik}, \dots, \beta_{mik})$ is the actual path of the unit i at time t_{ijk} (the times need not be the same for all units) and $\epsilon_{ijk} \sim N(0, \sigma_{\epsilon k})$ is a residual deviation for the unit i at time t_j under the stress level k. The total number of observations on unit i under stress k is denoted by n_{ik}. For the i^{th} unit under the stress k, $\beta_{1ik}, \dots, \beta_{mik}$ is a vector of m unknown parameters. A unit i under stress level k is assumed to fail when its degradation level first reaches to a predefined threshold level D_f.

The nonlinear mixed-effects degradation model [198] is used. The unit-to-unit variability in model parameters $\beta_{1k}, \dots, \beta_{mk}$ can be modeled with k multivariate normal distributions (one for each stress level) with mean vectors $\boldsymbol{\mu}_{\boldsymbol{\beta}_k}$ and covariance matrices $\boldsymbol{\Sigma}_{\boldsymbol{\beta}_k}$ [14]. It is generally assumed that the random parameters $\beta_{1k}, \dots, \beta_{mk}$ are independent of the ϵ_{ijk} and that $\sigma_{\epsilon k}$ is constant for each stress level k. Let $\boldsymbol{\theta}_{\boldsymbol{\beta}_k} = \left(\boldsymbol{\mu}_{\boldsymbol{\beta}_k}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}_k}\right)$ denote the overall population/process parameters for each stress level k.

The likelihood for the random-parameter degradation model is given as [14]:

$$L\left(\boldsymbol{\theta}_{\boldsymbol{\beta}_k}, \sigma_{\epsilon k} | DATA_k\right) = \prod_{i=1}^{p_k} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \left[\prod_{j=1}^{n_{ik}} \frac{1}{\sigma_{\epsilon k}} \varphi_{\text{nor}}\left(\zeta_{ijk}\right)\right] \times f_{\boldsymbol{\beta}_k}\left(\beta_{1ik}, \dots, \beta_{mik}; \boldsymbol{\theta}_{\boldsymbol{\beta}_k}\right) d\beta_{1ik}, \dots, \beta_{mik} \tag{2}$$

where $\zeta_{ijk} = \left[y_{ijk} - D(t_{ijk}, \beta_{1ik}, \dots, \beta_{mik})\right] / \sigma_{\epsilon k}$, and $f_{\boldsymbol{\beta}_k}\left(\beta_{1ik}, \dots, \beta_{mik}; \boldsymbol{\theta}_{\boldsymbol{\beta}_k}\right)$ is the multivariate normal distribution density function. The evaluation of equation (2) requires the numerical approximation of p_k integrals of dimension m (under multi-stress combination level , p_k is the number of sample paths and m is the number of parameters for each path). Maximizing equation (2) with respect to $(\boldsymbol{\mu}_{\boldsymbol{\beta}_{\boldsymbol{k}}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}_{\boldsymbol{k}}}, \sigma_{\epsilon k})$ to get the ML estimated parameters will be extremely difficult. Therefore, numerical methods are suggested.

Here, we assume that the degradation follows an exponential degradation path $D(t)$, which can be written in terms of random parameters $\beta_1, \beta_2,\ \beta_3$ as [191]:

$$D(t) = \beta_1 + \beta_2\left(e^{\beta_3 t} - 1\right) \tag{3}$$

where, β_1 represents the initial performance value and $\beta_2,\ \beta_3$ define the shape and the degradation rate that are assumed to be material and stress-dependent. The MLE estimates for each stress level $\widehat{\boldsymbol{\mu}}_{\boldsymbol{\beta}_{\boldsymbol{k}}}, \widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}_{\boldsymbol{k}}}$ can be estimated from the degradation data by applying the nonlinear mixed-effects model. In the degradation path defined in equation (3), the model parameter $\boldsymbol{\mu}_{\boldsymbol{\beta}_{\boldsymbol{k}}}$ is a three-dimension vector and $\boldsymbol{\Sigma}_{\boldsymbol{\beta}_{\boldsymbol{k}}}$ is a three-by-three covariance matrix. The exact ML estimates of model parameters $\boldsymbol{\theta}_{\boldsymbol{\beta}_{\boldsymbol{k}}} = (\boldsymbol{\mu}_{\boldsymbol{\beta}_{\boldsymbol{k}}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}_{\boldsymbol{k}}})$ given the data (degradation paths observations) at each stress level k would come from maximizing equation (2) with respect to $(\boldsymbol{\mu}_{\boldsymbol{\beta}_{\boldsymbol{k}}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}_{\boldsymbol{k}}}, \sigma_{\epsilon k})$. In this study, the 'nlme' R function [199] will be used instead to get approximate estimates for $\boldsymbol{\theta}_{\boldsymbol{\beta}_{\boldsymbol{k}}}$

Given a failure threshold D_f, the time at which a sample reaches the failure threshold D_f, is defined as lifetime L or time-to-failure, and is given by:

$$L = t_f = \frac{Ln\left(\frac{D_f - \beta_1}{\beta_2} + 1\right)}{\beta_3} \tag{4}$$

After the degradation modeling pseudo failure times at different stress levels are estimated with equation (4). Since the simultaneous application of multiple stress factors is considered, an equivalent stress model is proposed for developing a stress-life (S-L) curve. A probabilistic multi-stress dependent degradation model is formulated considering the S-L curve and the damage accumulation models. Finally, a multi-stress dependent dynamic reliability assessment model is proposed that is capable to provide reliability estimates for any given operating conditions. Although FHE degradation data are used to demonstrate the approach, the framework can be used for other kind of products, considering the appropriate degradation model, e.g., linear, nonlinear, asymptotic, etc.

3.3.1. Evaluation of time-to-failure distribution

The time at which a sample reaches the failure threshold (D_f) is defined as product lifetime (L) at a given stress level, which will vary from unit to unit and hence treated as a random variable. Generally, the distribution of product lifetime is defined by the distribution of degradation y_{ijk} [14]. Such distribution could be written in terms of the degradation model parameters. The sample degradation paths, failure threshold D_f, and lifetime distribution are graphically represented in Figure 3. Let us assume that a unit fails at time t if the degradation reaches D_f at time t, the probability of failure is given as:

$$Pr(L \leq t) = F(t) = F\left(t; \boldsymbol{\theta}_{\boldsymbol{\beta}_{\boldsymbol{k}}}\right) = \Pr\left[D(t, \beta_{1k}, \dots, \beta_{mk}) \geq D_f\right] \qquad (5)$$

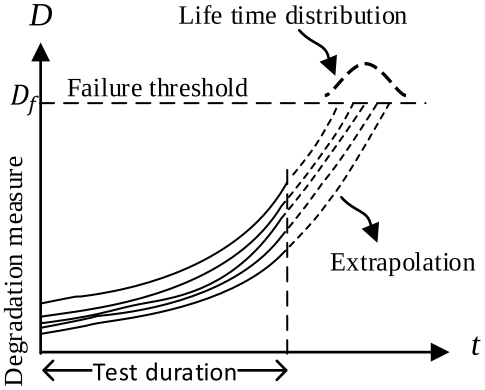


Figure 3. Degradation path and lifetime distribution

For a fixed D_f, the distribution of L depends on the distribution of $\beta_{1k}, \dots, \beta_{mk}$, that is, it depends on the basic path parameters in $\boldsymbol{\theta}_{\boldsymbol{\beta}_{\boldsymbol{k}}}$. For simple linear path models, $F(t)$ can be expressed as a function of the basic parameters in a closed form. However, for most practical models, specially nonlinear ones with random model parameters $\beta_{1k}, \dots, \beta_{mk}$, generally there is no closed-form solution for $F(t)$ [14]. These complex nonlinear models can only be evaluated with numerical methods. Therefore, we propose to use Monte Carlo simulation [14] approach as discussed below. This procedure needs to be repeated for each stress level k:

1. Generate N_k simulated realizations $\breve{\beta}_{1k}, \dots, \breve{\beta}_{mk}$ of $\beta_{1k}, \dots, \beta_{mk}$ from a multivariate normal distribution with mean $\hat{\boldsymbol{\mu}}_{\boldsymbol{\beta_k}}$ and variance-covariance matrix $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta_k}}$, where N_k is a large number.
2. Compute the N_k simulated failure times corresponding to the N_k realizations $\breve{\beta}_{1k}, \dots, \breve{\beta}_{mk}$. In general, this can be done by substituting the realizations $\breve{\beta}_{1k}, \dots, \breve{\beta}_{mk}$ into $D(t, \beta_{1k}, \dots, \beta_{mk})$, finding the crossing time $\breve{t}_{wk}$ for each realization (for $w = 1, \dots, N_k$). In the case study, we will use equation (4) to find the crossing time for each simulated path.
3. Do distribution analysis of the simulated crossing times to estimate a time-to-failure distribution, that is, use the simulated failure times $\breve{t}_{wk}$ to fit a distribution for the life, L_k. The parameters of the distributions can be estimated from the simulated failures times $\breve{t}_{wk}$

Although we provide a generic approach to identify the appropriate distribution of failure time estimated using degradation data, most of the existing literature assumes that L_k follows a normal distribution $N(\mu_{L_k}, \sigma_{L_k}^2)$ [200-202]. Therefore, we also assume that estimated failure time follows normal distribution. The MLE estimates of the parameters of time-to-failure distributions at each stress level k can be calculated as:

$$\hat{\mu}_{L_k} = \frac{1}{N_k} \sum_{w=1}^{N_k} \breve{t}_{wk} \tag{6}$$

$$\hat{\sigma}_{L_k}^2 = \frac{1}{N_k}\sum_{w=1}^{N_k}\left(\breve{t}_{wk} - \widehat{\mu_{L_k}}\right)^2 \tag{7}$$

3.3.1.1. S-L curve for multi-stress scenario

In damage theory, generally a single stress factor is considered as the loading factor such as fatigue load. For single stress factor, the relationship between fatigue life (L) and stress (S) is well explained by the S-L curve equation as given below [203, 204]:

$$L(S) = AS^{-\gamma} \tag{8}$$

The equation (8) actually represents the power law [190] that expresses the relationship between the product life and single accelerating stress factor as shown in Figure 4.

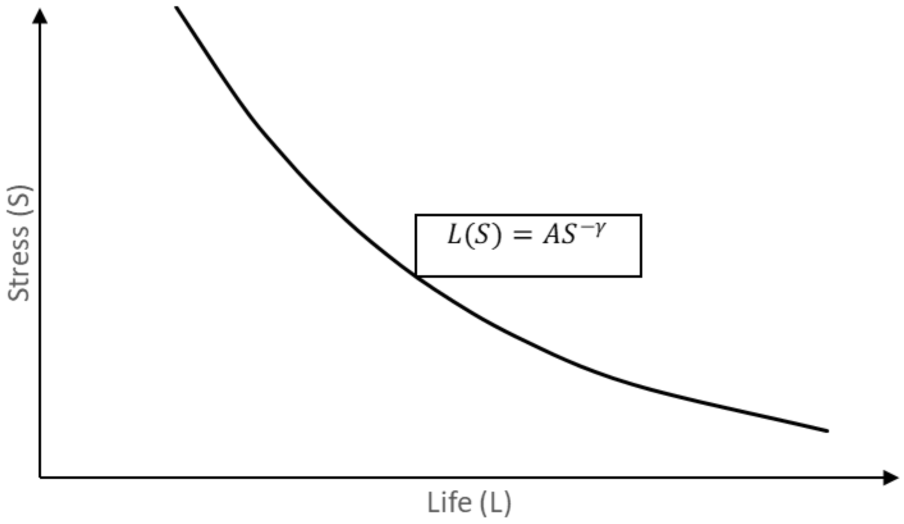


Figure 4. Generic stress-life (S-L) curve

However, most of the FHE products are exposed to a multi-stress factor environment in field operations. Therefore, to generate a (multi)S-L curve for reflecting real-life application, it seems reasonable to calculate an equivalent single stress to represent multi-stress conditions. For most of the electronic products temperature and humidity are the most commons stressors [205, 206]. Generally, the thermal stress can be modeled with the well-known physics of failure model called Arrhenius law and non-thermal stresses can be explained by the power-law model. In the presence of both thermal (*T*) and non-thermal stresses (U), the life-stress model can be expressed by the general Eyring law model as:

$$L = L(T, U) = A_0 U^{-\gamma_0} e^{\left(\frac{B_0}{T}\right)} \tag{9}$$

where A_0, B_0, and γ_0 are constants that need to be estimated using the experimental data. L is a specific characteristic of the life distribution [190]. For a normal distribution, it represents the mean life and therefore, equation (9) can be rewritten as:

$$\mu_L = A_0 \left(U e^{-\left(\frac{B_0}{T\gamma_0}\right)} \right)^{-\gamma_0} \tag{10}$$

Without loss of generality, let

$$S_{eq} = U e^{-\left(\frac{B_0}{T\gamma_0}\right)} \tag{11}$$

where S_{eq} can be considered an equivalent acceleration stress function, as long as it depends directly on the product of monotone functions of thermal (*T*) and non-thermal (*U*) stresses [207, 208]. Non-thermal stress (*U*) could be any other single stress variable, such as humidity, voltage, current. It could also represent multiple non-thermal stresses; where *U* could be defined as a function of (possibly) transformed stresses [190, 207, 208]. For instance, considering a generalized power-law model, q non-thermal stresses U_v, can generate equivalent non-thermal accelerating stress *U* [207]:

$$U = \prod_{v=1}^{q} \left(F_m(U_v) \right)^{\eta_v} \tag{12}$$

where $F_m(\cdot)$ represents any monotone function.

By combining equations (10) and (11), we derive an equivalent stress and life model:

$$\mu_L = A_0 S_{eq}^{-\gamma_0} \tag{13}$$

The life-stress model given by equation (13) is similar to equation (8) with the advantage that equation (13) captures multi-stress scenario in the form of equivalent stress S_{eq}.

In our study, we consider temperature and humidity as the (two) stress factors. Considering the non-thermal stress (U) to be the relative humidity H and assuming $q = 1, F_m(U_v) = U_v, \eta_v = 1$ in equation (12), the equation (11) can be modified to calculate equivalent stress as:

$$S_{eq} = He^{-\left(\frac{B_0}{T\gamma_0}\right)} \tag{14}$$

The expected life or time to failure μ_L is given as:

$$\mu_L = A_0\left(He^{-\left(\frac{B_0}{T\gamma_0}\right)}\right)^{-\gamma_0} = \frac{A_0}{H^{\gamma_0}}e^{\frac{B_0}{T}} \tag{15}$$

The equation (15), in this particular case, turns out to be the well-known Eyring model.

The parameters A_0, B_0, γ_0 in equation (15) can be estimated by using the least square method for multiple linear regression.

By taking the natural log, the equation (15) becomes:

$$Ln(\mu_L) = Ln(A_0) - \gamma_0 Ln(H) + B_0.\frac{1}{T} \tag{16}$$

The following transformations are considered to linearize the model:

$$Ln(\mu_L) = y', Ln(H) = x_1, \frac{1}{T} = x_2, Ln(A_0) = \theta'_0, -\gamma_0 = \theta'_1, B_0 = \theta'_2 \tag{17}$$

The notation y' is used here to avoid confusion with the observed sample degradation path y_{ijk} in equation (1). We get a linearized model:

$$y' = \theta'_0 + \theta'_1 x_1 + \theta'_2 x_2 \tag{18}$$

The parameters θ'_0, θ'_1, and θ'_2 can be estimated from the (linearized) life – stress points (y'_k, x_{k1}, x_{k2}), where:

$$y_k' = Ln\left(\hat{\mu}_{L_k}\right), x_{k1} = Ln(H_k), x_{k2} = \frac{1}{T_k} \tag{19}$$

There is one life – stress point for each stress level k, where $k = 1,2, \dots, r$. For each stress level k with specified temperature and humidity (T_k, H_k) there is an associated expected failure time, $\hat{\mu}_{L_k}$ estimated using equation (4) . The multiple linear model can be written as a set of r equations:

$$y_k' = \theta'_0 + \theta'_1 x_{k1} + \theta'_2 x_{k2} + \epsilon_k, \text{for all } k = 1,2, \dots, r \tag{20}$$

Using matrix notation, the model for r life-stress points can be written as

$$\begin{pmatrix} y_1{}' \\ y_2{}' \\ \vdots \\ y_r{}' \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ \vdots & \vdots & \vdots \\ 1 & x_{r1} & x_{r2} \end{pmatrix} \begin{pmatrix} \theta_{\prime 0} \\ \theta'_1 \\ \theta'_2 \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_r \end{pmatrix} = \theta'_0 + \theta'_1 x_1 + \theta'_2 x_2 \tag{21}$$

Or in a condensed form:

$$\boldsymbol{y} = \boldsymbol{X\theta'} + \boldsymbol{\epsilon} \tag{22}$$

For the multiple linear model, we consider the following assumptions:

- $E(\epsilon_k) = 0$ for all $k = 1,2, \dots, r$
- $var(\epsilon_k) = \sigma^2$ for all $k = 1,2, \dots, r$
- $cov(\epsilon_k, \epsilon_l) = 0$ for all $k \neq l$

Provided the multiple linear model assumptions are valid, the least-squares estimates of $\widehat{\boldsymbol{\theta'}} = \left(\widehat{\theta'}_0, \widehat{\theta'}_1, \widehat{\theta'}_2\right)^{\boldsymbol{T}}$ is given as [209]:

$$\widehat{\boldsymbol{\theta'}} = (\boldsymbol{X^T X})^{-1} \boldsymbol{X^T y'} \tag{23}$$

where $(\cdot)^{\boldsymbol{T}}$ represents the transpose operator.

By applying the inverse transformations assumed, we can get the estimates of parameters (A_0, B_0, γ_0) corresponding to the original nonlinear model in equation (15).

$$\widehat{A_0} = e^{\widehat{\theta'}_0}, \widehat{B_0} = \widehat{\theta'}_2, \text{and } \widehat{\gamma_0} = -\widehat{\theta'}_1 \tag{24}$$

A similar approach can be followed to estimate the parameters for other non-thermal stress factors or multiple non-thermal stresses (U) combined with temperature (T).

The estimated parameters can now be used to calculate the equivalent stress (S_{eq}) using equation (14). Once the equivalent stress is calculated for each multi-stress combination, the S-L curve for FHE products can be fitted using pairwise points *equivalent stress - expected time to failure* ($S_{eqk}, \hat{\mu}_{Lk}$).

In the next section, we develop a probabilistic degradation model using a fitted S-L curve for FHE products and Palmgren-Miner's linear damage accumulation model.

3.3.2. Probabilistic damage accumulation model

The damage accumulation is a cumulative deteriorating phenomenon by which the product reaches to its failure state gradually. Since, the damage accumulation is influenced by several random variables such as operating conditions, material type, and usage rate, it also behaves as a random variable. Considering the probabilistic nature of damage accumulation, the distribution function of damage accumulation can be treated as a normal distribution [202]:

$$D'^{(t)} \sim N\{\mu_{D'}(t), \sigma^2_{D'}(t)\} \tag{25}$$

where *D'(t)* is the accumulated damage with mean $\mu_{D'}(t)$ and variance $\sigma^2_{D'}(t)$. We use $D'(t)$ to represent the damage accumulation to avoid confusion with the degradation path $D(t)$.

The Palmgren-Mine's linear damage accumulation model defines the damage as a ratio of the number of usage cycles to total expected life cycles at a given stress level. Nevertheless, under constant stressors a ratio of usage time to expected lifetime for a given stress level (or multi-stress combination) is more appropriate to consider. Assuming no initial damage, the expected damage accumulation $\mu_{D'_k}$ at a given stress combination (e.g. combination of temperature and humidity) level k can be calculated as [204]:

$$\mu_{D'_k} = \frac{t_k}{\mu_{L_k}} \tag{26}$$

where, t_k is the usage time and μ_{L_k} is the expected time to failure at any given combined single stress level k. If the product is subjected to n multi-stress levels (a sequence of n single stress levels), total expected damage accumulation is given as $\mu_{D'}$:

$$\mu_{D'} = \sum_{k=1}^{n} \mu_{D'_k} = \sum_{k=1}^{n} \frac{t_k}{\mu_{L_k}} \tag{27}$$

where $\mu_{D'}$ is the total expected damage accumulation, $\mu_{D'_k}$ is the expected damage accumulation when subjected to *k^th^* stress level combination, t_k is the operation or usage time at stress combination level k**.** μ_{L_k} is the expected time to failure at the *k^th^* stress level combination and can be estimated from the degradation model, equation (6).

Combining the life-stress model equation (13) and damage accumulation model given in equation (26), the expected damage accumulation, $\mu_{D'_k}$, can be written as:

$$\mu_{D'_k} = \frac{{S_{eq_k}}^{\gamma_0}}{A_0} t_k \tag{28}$$

Considering multiple levels of multi-stress combinations, the total expected damage accumulation is given as:

$$\mu_{D'} = \sum_{k=1}^{n} \mu_{D'_k} = \sum_{k=1}^{n} \frac{{S_{eq_k}}^{\gamma_0}}{A_0} t_k \tag{29}$$

where S_{eq_k} can be calculated for each stress level combination k by using equation (11). For the given stress factors temperature T_k and humidity H_k, S_{eq_k} can be calculated by using equation (14) (letting $T = T_k, H = H_k$). The equation (29) is modified by replacing S_{eq_k} and given as:

$$\mu_{D'} = \sum_{k=1}^{n} \frac{H_k^{\gamma_0}}{A_0 e^{\left(\frac{B_0}{T_k}\right)}} t_k \tag{30}$$

It is generally assumed that a failure occurs when total accumulated damage D' reaches to the threshold damage (D'_c). Considering threshold damage also a random variable, the expected value it is $E(D'_c) = \mu_{D'_C} = 1$.

Since we have assumed that the damage accumulation follows a normal distribution $NOR(\mu_{D'_k}, \sigma^2_{D'_k})$, the standard deviation of accumulated damage at time t_k is given as [210]:

$$\sigma_{D'_k} = \frac{{S_{eq_k}}^{\gamma_0}}{A_0} \left(\frac{\sigma_{L_k}}{\mu_{L_k}} \right) t_k \tag{31}$$

where, μ_{L_k} and σ_{L_k} are the mean and standard deviation of the product life L_k at combined stress level *k*, respectively.

Considering random variability in operating conditions, it is assumed these FHE products will be subjected to multiple levels of multi-stress combination during field operations. Given multi-stress factors and multi-level scenario, the standard deviation in the accumulated damage can be calculated as [210]:

$$\sigma_{D'} = \sqrt{\sum_{k=1}^{n} \left(\frac{{S_{eq_k}}^{\gamma_0}}{A_0} t_k \left(\frac{\sigma_{L_k}}{\mu_{L_k}} \right) \right)^2} \tag{32}$$

Equation (32) is modified by replacing S_{eq_k} and given as:

$$\sigma_{D'} = \sqrt{\sum_{k=1}^{n} \left(\frac{H_k^{\gamma_0}}{A_0 e^{\left(\frac{B_0}{T_k}\right)}} t_k \left(\frac{\sigma_{L_k}}{\mu_{L_k}} \right) \right)^2} \tag{33}$$

Both the expected damage accumulation given by equation (30) and the standard deviation of damage accumulation given by equation (33) are functions of the stress levels, time duration the product is subjected to these stress levels, the expected product life, and variability (or standard deviation) in expected product life at any given stress level. Essentially, these two equations represent dynamic functions and provide expected accumulated damage and its variability at any given time when a product is subjected to multiple stresses at different levels during field operation or testing.

3.3.3. Dynamic reliability estimation

After deriving dynamic functions of expected damage accumulation and its variability, we now use stress-strength interference model to develop a dynamic reliability assessment model. In order to use stress-strength interference model, we consider the following assumptions [210]:

- Failure occurs when the damage accumulation (D') reaches the threshold damage (D'_c), where $E(D'_c) = \mu_{D'_C} = 1$
- The threshold damage has the same distribution as the damage accumulation measure
- When the usage life is equal to the mean failure life ($t = \mu_L$), the variability of threshold damage accumulation $\left(\sigma_{D'_c}{}^2\right)$ is equal to the variability of the damage accumulation measure $(\sigma_{D'}{}^2)$

In the stress-strength interference model, the reliability is given as the probability that the accumulated damage is less than the threshold damage:

$$R = P\{D' < D'_c\} \tag{34}$$

$$R = 1 - P\{D'_c - D' \leq 0\} \tag{35}$$

$$R = 1 - \Phi\left(-\frac{\mu_{D'_c} - \mu_{D'}}{\sqrt{\sigma_{D'_c}{}^2 + \sigma_{D'}{}^2}}\right) \tag{36}$$

where, Φ is the cumulative density function of the standard normal distribution, and $\mu_{D'}$ is total accumulated damage at a given time.

In our third assumption, we considered that at failure time (when usage time is equal to product life) in both the damage accumulated and the threshold damage the variability is the same. Hence, the variability in threshold damage can be calculated by considering usage time $t_k = \mu_{L_k}$ in equation (32).

Figure 5 shows the damage accumulation probability density (pdf) function (black lines) is moving up (towards the threshold damage line) with time whereas pdf of threshold damage (red line) remains stationary. It is also important to note that for damage accumulation pdf, both the expected accumulated damage and its variability are changing (increasing) with usage time. Because of this phenomenon, we call it a dynamic reliability assessment model because it captures the impact of both usage time and field operating conditions on both expected damage accumulation and its variability.

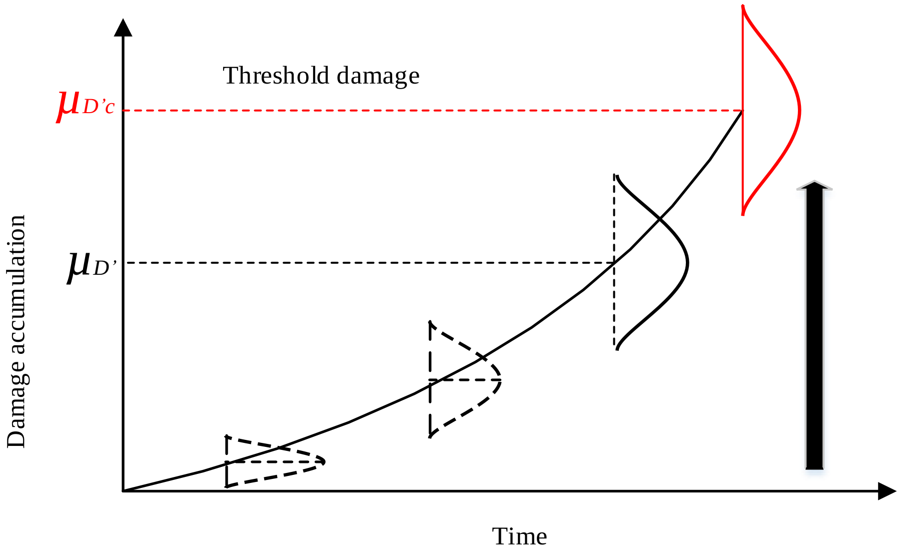


Figure 5. Dynamic stress-strength model of damage accumulation

Let us assume that the mean threshold damage $\mu_{D'_c} = 1$ and considering appropriate equations for damage accumulation and its variability, equation (36) can be rewritten as:

$$R = 1 - \Phi\left(- \frac{1 - \sum_{k=1}^{n} \frac{{S_{eq_k}}^{\gamma_0}}{A_0} t_k}{\sqrt{{\sigma_{D'_c}}^2 + \sum_{k=1}^{n} \left(\frac{{S_{eq_k}}^{\gamma_0}}{A_0} t_k \left(\frac{\sigma_{L_k}}{\mu_{L_k}} \right) \right)^2}} \right) \tag{37}$$

The equation (37) is modified by replacing S_{eq_k} and given as:

$$R = 1 - \Phi\left(- \frac{1 - \sum_{k=1}^{n} \frac{H_k^{\gamma_0}}{A_0 e^{\left(\frac{B_0}{T_k}\right)}} t_k}{\sqrt{{\sigma_{D'_c}}^2 + \sum_{k=1}^{n} \left(\frac{H_k^{\gamma_0}}{A_0 e^{\left(\frac{B_0}{T_k}\right)}} t_k \left(\frac{\sigma_{L_k}}{\mu_{L_k}} \right) \right)^2}} \right) \tag{38}$$

If the given product is subjected to a single stress level, the equation (29) can be evaluated considering $n = 1$ to estimate the reliability.

To estimate the reliability under given stress (operating) conditions by using equation (38), both the expected time to failure (μ_L) and its standard deviation (σ_L) are needed. By substituting the estimated parameters of the life stress model, e.g., (A_0, B_0, γ_0) in equation (15), it is possible to estimate the expected time to failure (μ_L) under given stress conditions. To predict the reliability, it is also needed the standard deviation (σ_L) of the expected time-to-failure under given stress conditions, which is unknown. Further, many existing models such as Arrhenius-lognormal or power-Weibull have assumed the constant spread of time-to-failure distribution [190]. However, recent literature shows that the spread of time-to-failure is a function of stress [190] and usually the spread or variability in failure time is higher at a lower stress level. At lower stress levels, it is expected to observe larger time-to-failure (longer life of the product) but higher variability in the life of the product. Therefore, it is proposed to use a linear model to estimate the standard deviation of time-to-failure as a function of the mean time-to-failure, as shown in Figure 6. By setting the intercept term to zero, which assures that the estimated standard deviation is positive as time-to-failure is positive, the standard deviation function is given as:

$$\sigma_L = \delta\mu_L \tag{39}$$

where δ is a characteristic of the product and represents the slope of the standard deviation as a function of the expected product life. This linear relationship is illustrated in Figure 6.

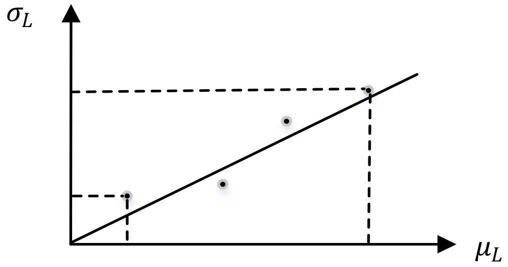


Figure 6. The standard deviation of expected product life as a function of product life.

The parameter δ can be estimated from the pairwise points $\left(\hat{\sigma}_{L_k}, \hat{\mu}_{L_k}\right)$, represented as dots in Figure 6. Each pair of observation at each stress level k, where $k = 1,2, \dots, r$, consists of expected product life $\hat{\mu}_{L_k}$ (equation (6)) and variability in product life $\hat{\sigma}_{L_k}$(equation (7)). We propose to use the linear regression model to capture the relationship between $\hat{\sigma}_{L_k}$ and $\hat{\mu}_{L_k}$:

$\sigma_{L_k} = \delta\mu_{L_k} + \epsilon_k$, for all $k = 1,2, \dots, r$

The least-squares estimator $\hat{\delta}$ is then given by [211]:

$$\hat{\delta} = \frac{\sum_{k=1}^{r} \hat{\mu}_{L_k}\hat{\sigma}_{L_k}}{\sum_{k=1}^{r} \hat{\mu}_{L_k}^2} \tag{40}$$

By substituting $\hat{\delta}$ from equation (40) in equation (39), the estimated standard deviation $\widehat{\sigma_L}$ of failure time at any condition is given as:

$$\widehat{\sigma_L} = \hat{\delta}\mu_L \tag{41}$$

3.3.4. Case study

To demonstrate the proposed method, a flexible hybrid electronics (FHE) product has been considered. The FHE is the newly developed complex concept that has immense opportunities for applications, especially in next-generation electronic products, security, defense, and wearable technology. Generally, a FHE device consists of a plastic substrate, an electric circuit, and, in some cases, an adhesive layer between the substrate and the circuit. The FHE samples used in our case study have been fabricated on a flexible polyamide substrate using a thinned (25µm) bare silicon die bonded to the traces on the substrate using an anisotropic conductive paste (ACP). Figure 7 shows the FHE test sample used in this work.

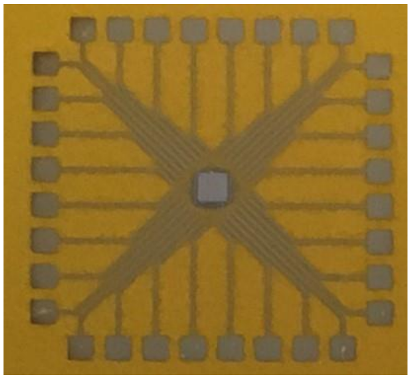


Figure 7. The flexible hybrid electronics device used in this work

To get the time-to-failure information on FHE samples, ADT has been conducted using temperature and humidity stress factors simultaneously. A common performance indicator for FHE has been daisy-chain resistance [159, 165-169, 181]. It is well known that the major failure mechanisms for FHE's are humidity absorption of the substrate and the adhesive, and delamination in substrate-circuit joint [159, 165-169]. In ADT, four treatment combinations (or levels) of two stress factors (each at two levels) have been considered to get the stress effect on failure mechanism. At each treatment combination, five samples were allocated for testing and measurement. The changes in resistance as an indicator of performance measurement have been observed and recorded at every five hours. A sample is considered to have failed when there is a 20% increase in the initial nominal resistance [186].

3.3.4.1. Results

Figure 8 shows the degradation path (dashed lines) for the five samples at treatment combination level 2 (85 °C temperature and 85% relative humidity (RH). The predicted population path (solid line) is fitted by using ML estimates $\hat{\boldsymbol{\mu}}_{\boldsymbol{\beta}_2}$, which seems to be a good fit to the degradation data.

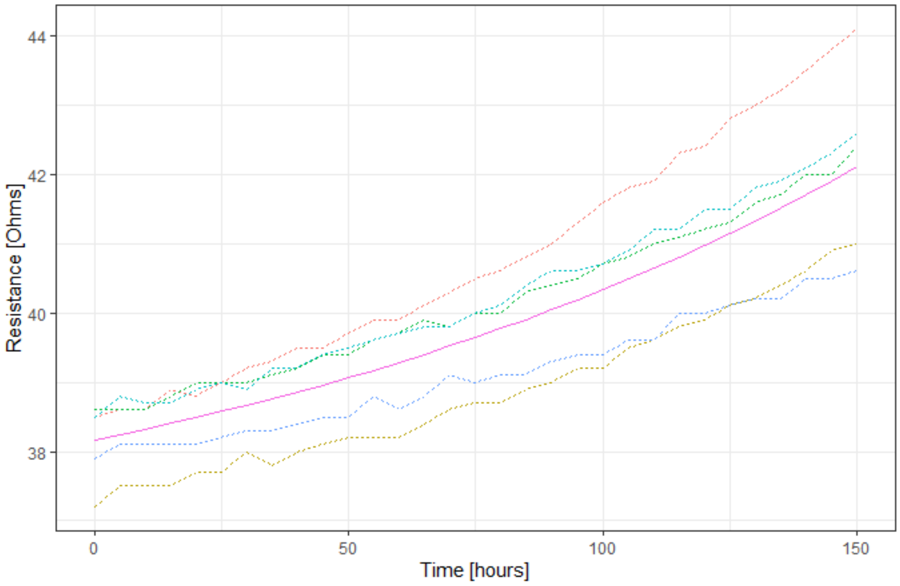


Figure 8. Degradation paths (dashed lines) and fitted path at temperature 85 °C and RH 85%

The ML estimates $\boldsymbol{\mu}_{\boldsymbol{\beta}_k}$and $\boldsymbol{\Sigma}_{\boldsymbol{\beta}_k}$, at each treatment combination were computed using *nmle* function in the R package. The results are shown in Table 2.

Table 2. ML estimates

Stress level	$\boldsymbol{\mu}_{\boldsymbol{\beta}_{\boldsymbol{k}}}$	$\boldsymbol{\Sigma}_{\boldsymbol{\beta}_{\boldsymbol{k}}}$
85 °C/98% RH $k = 1$	$\begin{pmatrix} 38.47 \\ 2.58 \\ 9.68e-03 \end{pmatrix}$	$\begin{pmatrix} 0.22637 & 0.08447 & 1.22e-04 \\ 0.08447 & 0.05428 & 1.74e-04 \\ 0.00012 & 0.00017 & 7.93e-07 \end{pmatrix}$
85 °C/85% RH $k = 2$	$\begin{pmatrix} 38.17 \\ 2.19 \\ 6.86e-03 \end{pmatrix}$	$\begin{pmatrix} 0.21862 & 0.16457 & -1.38e-04 \\ 0.16457 & 0.12951 & -4.72e-05 \\ -1.38e-04 & -4.72e-05 & 6.55e-07 \end{pmatrix}$
65 °C/98% RH $k = 3$	$\begin{pmatrix} 37.56 \\ 2.65 \\ 2.70e-03 \end{pmatrix}$	$\begin{pmatrix} 2.26e-01 & 1.13e-01 & -7.96e-12 \\ 1.13e-01 & 2.67e-01 & -8.66e-12 \\ -7.96e-12 & -8.66e-12 & 2.81e-16 \end{pmatrix}$
65 °C/85% RH $k = 4$	$\begin{pmatrix} 37.45 \\ 3.28 \\ 1.29e-03 \end{pmatrix}$	$\begin{pmatrix} 1.09183 & -1.24e-03 & 3.07e-05 \\ -1.24e-03 & 7.87e-06 & -3.71e-08 \\ 3.07e-05 & -3.71e-08 & 8.67e-10 \end{pmatrix}$

The normalized equivalent stress $S_{eqk,norm}$ is shown in Figure 9.

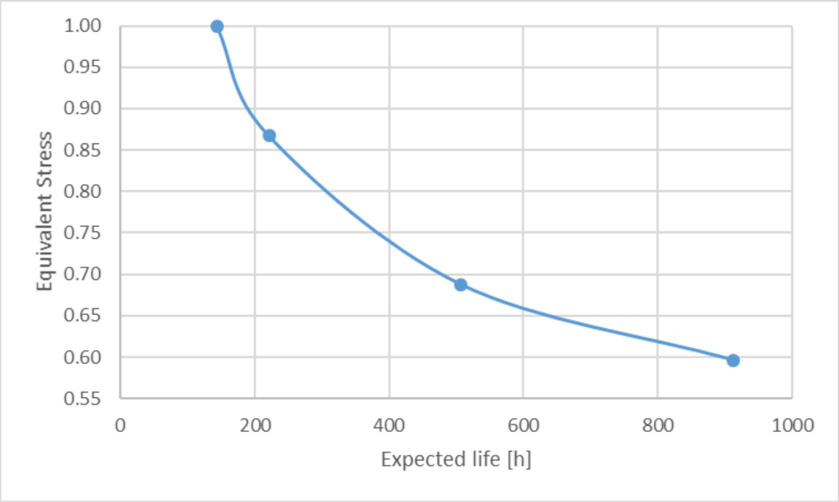


Figure 9. Stress-life curve for FHE product considering equivalent stress

Figure 10 shows the reliability estimated by the model for different constant operating conditions.

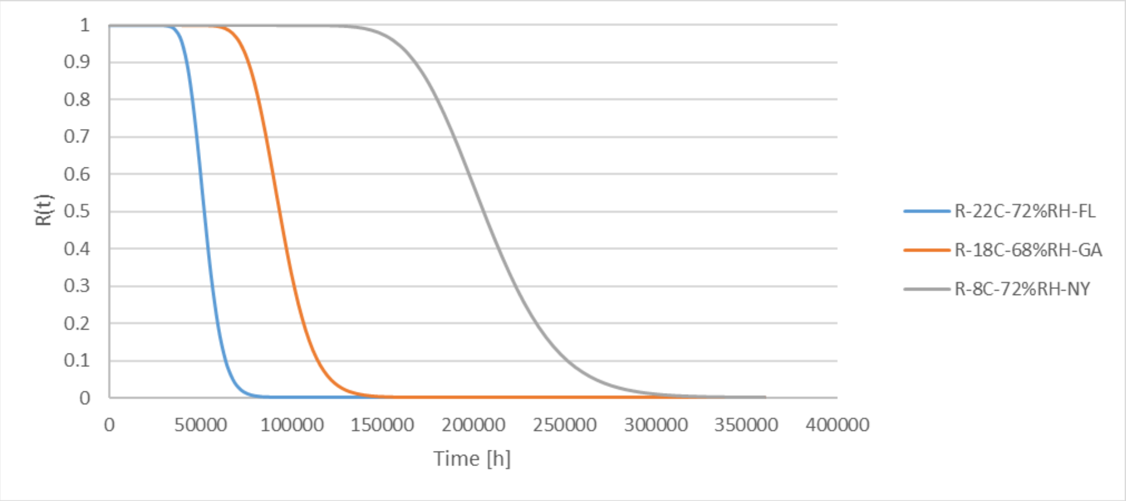


Figure 10. Reliability at different normal conditions scenarios

Most of the products operate in multi-stress scenarios at different levels of stress. Further, the order or sequence of changes in the operating condition also influences product reliability [210]. To demonstrate the impact of sequencing in which operating conditions change or the product undergoes, we consider two sequencings of three multi-stress scenarios:

Sequence 1: 22C/72%RH (average conditions of Florida, FL), 18C/68%RH (average conditions of Georgia, GA), and 8C/72%RH (average conditions of New York, NY) for 10,000 hours, 15,000 hours, and the remaining hours up to failure, respectively.

Sequence 2: the second sequencing is obtained by reversing the order to NY-GA-FL for 133,438, 15,000, and 10,000 hours, respectively. The reliability assessment provided by the proposed model of both sequencing conditions is given in Figure 11.

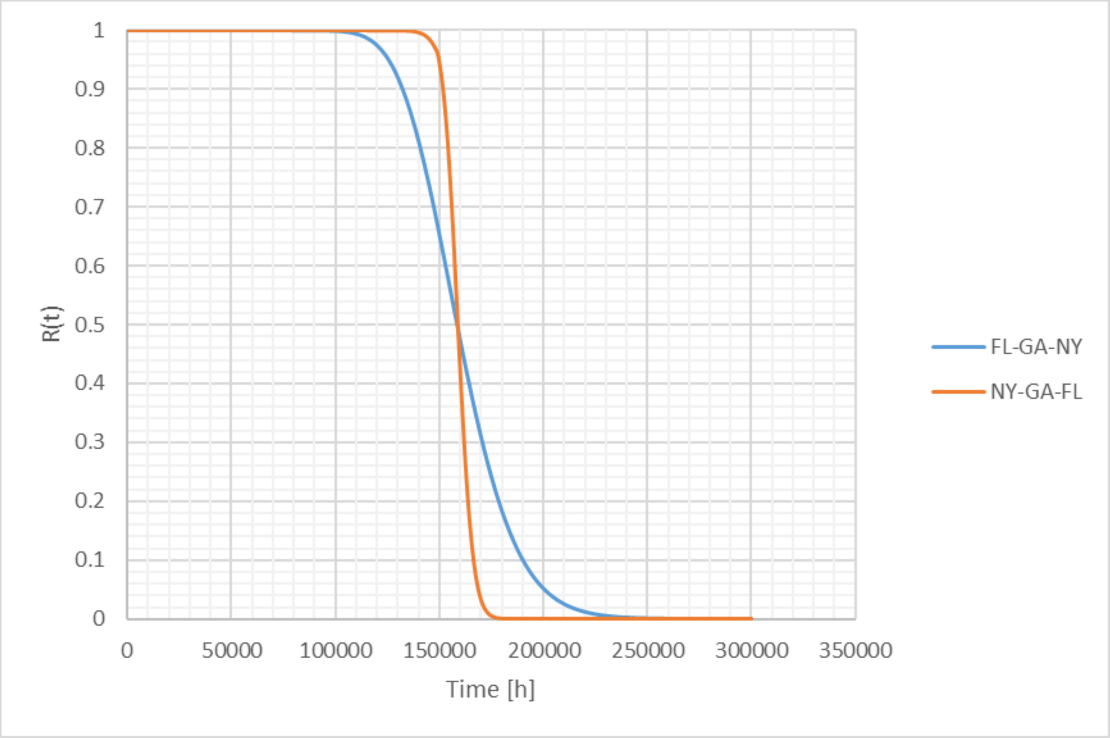


Figure 11. Reliability estimation under variable operating conditions

Figure 11 shows that for NY-GA-FL sequence, the initiation period is longer (almost constant reliability) but after the failure process starts, the reliability falls almost vertically, and the product fails within very short span of time. This behavior can be explained by the intensity of operating stress levels and their order. The intensity of operating conditions in NY area is low (8C/72%RH) and therefore, it takes longer to initiate the failure mechanism. However, once the failure process is initiated and the product is subjected to more intense operating conditions such as FL (22C/72%RH), the failure process propagates at a faster rate causing an almost vertical drop in reliability. On the other hand, when the product is subjected to severe operating conditions (22C/72%RH-FL) initially, the failure process initiates earlier but failure propagation slows down when operating conditions change from severe to medium and low. Because of slower propagation of the failure process, the degradation process slows down causing slight improvement in product reliability as can be shown in Figure 10. This explains FHE products are not only vulnerable to varying operating conditions but to the pattern in which operating conditions change also impacts the product performance and reliability.

3.3.5. Conclusion

This study proposed a framework based on degradation test data to estimate the reliability of FHE systems. It contemplates multi-stress factors combinations experiments whose resultant degradation data are used to fit a nonlinear mixed-effects regression model and subsequently, to numerically evaluate time-to-failure distributions. Furthermore, the mean-time-to-failure estimates for different stress combinations are then used to fit a developed multi-stress factor stress-life (S-L) model. The S-L model, based on equivalent stress, allows estimating the product-life under given operating conditions other than those from the original experiments. In addition, the framework suggests a method to estimate both the expected damage accumulation and its variability at any given time and operating conditions. Additionally, the framework provides a procedure to evaluate the reliability as a function of time and operating conditions, useful for both constant and sequences of (varying) operating conditions

CHAPTER 4. RELIABILITY ASSESSMENT OF CPS CONSIDERING HARDWARE-SOFTWARE INTERACTIONS [73, 74][2]

4.1. Abstract

Besides physical components reliability, an important aspect of complex CPS systems reliability is the interaction between hardware and software. Most of the existing work has assumed either independence between hardware (HW) and software (SW) or a fixed proportion of hardware reported failures to represent HW/SW interactions. These assumptions do not necessarily reflect reality. This chapter proposes a reliability model for a system that captures the changing interactions between hardware and software based on probabilistic models. A preliminary model that integrates hardware and software, as independent blocks, for reliability estimation of cyber-physical systems was presented in our paper published in RAMS 2019 [73]. In addition, a more elaborated approach, that incorporates probabilistic hardware/software interactions, is discussed in our paper submitted to *IEEE Transactions on Systems, Man, and Cybernetics: Systems* [74]. The remaining sections of this chapter are based on such paper.

[2] The present chapter is based on the following papers:

1. A. Davila-Frias, N. Yodo, and O. Yadav, "Probabilistic modeling of hardware and software interactions for reliability prediction of embedded systems," 2019. In IEEE Transactions on Systems, Man, and Cybernetics: Systems. Submitted.
   Contribution of Alex Davila Frias: developing the mathematical models, analysis of the case study, discussion of the results, and drafting the paper.
   Contribution of Nita Yodo and Om Yadav: verification of the results and proofreading the draft paper.

2. A. Davila Frias, N. Yodo, and O. P. Yadav, "Mixed-Degradation Profiles Assessment of Critical Components in Cyber-Physical Systems,". Published in 2019 Annual Reliability and Maintainability Symposium (RAMS), 28-31 Jan. 2019, pp. 1-6.
   Contribution of Alex Davila Frias: developing the mathematical models, analysis of the case study, discussion of the results, and drafting the paper.
   Contribution of Nita Yodo and Om Yadav: verification of the results and proofreading the draft paper.

4.2. Introduction

Hardware-software systems are present in many of today's CPS applications, such as cellphones, automobiles, military systems, medical equipment, and tracking devices. The reliability of hardware-software systems is a major concern because human lives and/or critical assets often depend on their proper function. Although extensive research has been done in hardware and software separately, the literature in capturing the impact of hardware-software (*HW-SW*) interactions for reliability modeling is still scarce and is summarized in the next paragraph.

Most of the existing research on hardware-software systems reliability assessment has assumed independence between hardware and software [25, 73, 123, 212, 213]. Another limitation most of the models face is the shortage of sufficient data on failures of HW/SW systems because such data are sensitive [27]. In addition, many of the common references had used data from the 1980s and 1990s, when computer systems were significantly different from today [124, 214-216]. A more recent study [217] analyzed a population of 100,000 disks in terms of the mean time to hardware failure without considering the interaction with the software. With the availability of a large dataset (23,000 failures over nine years period including 24,101 processors) provided by Los Alamos National Laboratory (LANL), Schroeder et al. [27] studied the mean failure times and the root cause of failures considering both hardware and software, separately though. Indeed, they reported that 53% of the failures were attributed by hardware and 22% by software. Based on the same dataset, El-Sayed et al. [218] studied the correlations between failures, including hardware, software, among others. Although they provided intuitive insights, their work did not focus on modeling the HW/SW interactions.

On the other hand, some works have considered HW/SW interactions. For example, a unified hardware-software reliability model was introduced by Welke et al. [123]. They considered the time-varying software failure intensity of the Goel-Okumoto/ non-homogeneous Poisson process (NHPP) model [119] and a Markov hardware reliability model. Recently, Teng et al. [22] and Roy et al. [23] proposed combined reliability models considering failures due to hardware and software interactions. HW/SW interactions in systems have been usually analyzed by assuming that a fixed percentage of observed "hardware" failures are, in reality, HW/SW interactions failures [22-25]. This might not represent the reality, because different percentages in the range from 20% to 35% have been reported as the actual HW/SW interactions proportion [26, 124, 219, 220]. Therefore, the fraction of HW/SW interaction failures in a new system is essentially unknown and random in nature, which cannot be neglected for reliability assessment. Therefore, an approach that captures the probabilistic nature of interactions is proposed to model and predict a hardware-software system's reliability.

Systems failure databases usually provide failure data classified as purely hardware and software failures. Nevertheless, a fraction of misclassified "hardware" failures can indeed be HW/SW interaction failures because degraded states of hardware may lead to software misbehavior and, eventually, system failure. Besides the traditional hardware and software reliability models, HW/SW interactions element is integrated into the system's reliability prediction model to provide a more realistic assessment in this study. This is achieved by considering the fraction of hardware failures, which are, in reality, HW/SW interaction failures, as a random variable. The randomness of the fraction of HW/SW interaction failures demands the use of stochastic programming methods to estimate the parameters of a hardware failure time distribution and the corresponding hardware reliability. To consider randomness, a stochastic optimization problem is formulated and solved with the general algebraic modeling system (GAMS) [221] software by defining an extended mathematical programming (EMP) model [222]. Software reliability is modeled with a non-homogeneous Poisson process (NHPP) to fit the software failure data [120, 223]. To capture the interaction between hardware and software, a Markov process model is used by considering that a degraded hardware state leads to a HW/SW failure and eventually to a system failure[22].

The estimation of rate parameters for the model becomes a challenging task because, first, the failure databases do not usually provide information on state transition rates, and second, the rate parameters depend on the random fraction of HW/SW interactions. To tackle this problem of estimating the rate parameter, a Monte Carlo method is employed in this work. Different from previous works, the ratio between state transition rates and the fraction of HW/SW interaction failures are treated as random variables and sampled according to assumed probability distributions with the aim to generate sampled distributions for the rate parameters. Different scenarios are considered by assuming several distributions for the HW/SW interactions and a uniform distribution for the state transition rates ratio. The sampled rate parameters are then used to generate sampled distributions for the reliability of the HW/SW element, which is then used along with pure hardware and pure software reliability estimates to generate sampled distributions of system reliability. By considering this probabilistic approach, confidence intervals and quartiles can be obtained for system reliability in addition to point estimators. The proposed methodology constitutes a practical tool and provides valuable information for designing better maintenance/warranty policies and developing availability models. The proposed method is demonstrated with data from a real computing system provided by Los Alamos National Laboratory [224].

4.3. Proposed reliability estimation methodology

Reliability is the probability of a system, subsystem, or component "to perform a required function under specified environmental and operational conditions for a stated period of time" [79-81]. Many modern systems, ranging from embedded systems to cyber-physical systems (CPSs), integrate hardware and software components. In these systems, hardware components may fail due to temperature, humidity, vibration, voltage, etc., causing a failure of the entire system. A hardware failure occurs when the hardware component ceases to function, e.g., the CPU burns out. On the other hand, a software failure may cause a system's failure as well, e.g., in an embedded system, the software can corrupt the operating system and crash the entire system [225, 226]. A software failure, defined as the occurrence of an "incorrect output as a result of an input value that is received with respect to the specification" [120], is caused by a fault triggered by a specific input. A fault is a manifestation of an error made by the programmer or designer, as shown in Figure 12 [227]. Therefore, in general, it has been assumed that a hardware-software system may fail due to: (1) pure hardware failures and (2) pure software failures. This work proposes to incorporate a third important element, i.e., (3) HW/SW interaction failures. The HW/SW interaction failures may be defined as the result of a change in the hardware characteristic leading to software operation under unexpected operational profiles [22, 228, 229]. For instance, in an embedded system, a degraded circuit component may cause a microcontroller to receive an input out of the logic voltage levels, which might result in unexpected firmware behavior and, eventually, a software failure. Similarly, a degraded reference chip may lead to wrong inputs connected to an analog-to-digital converter (ADC) input of a microcontroller running firmware. Also, in a computer system, a failure in the random-access memory (RAM) or permanent storage blocks can lead to a failure in an entire computer system. In the remainder of this chapter, this kind of failure will be regarded as HW/SW interaction failures. In the proposed approach, it is assumed that the software does not degrade [120]. Therefore, the possibility of a change in the software characteristic leading to hardware failure is not considered in the proposed model.

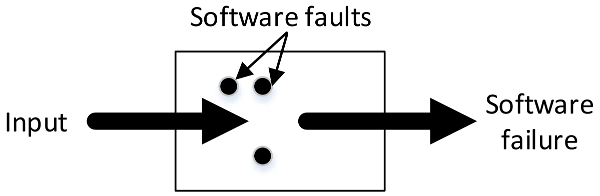


Figure 12. Software failure representation [120]

A hardware-software system can be represented as the integration of hardware, software, and HW/SW interactions (interface), as shown in Figure 13.

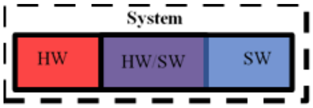


Figure 13. Representation of a hardware-software system with HW/SW interactions

Assumptions: (1) It is assumed that the system fails whenever a pure hardware failure, a pure software failure, or a HW/SW interaction failure occurs. (2) Furthermore, it is assumed that these failure types are mutually independent. (3) For hardware model, the failures can be modeled with a NHPP Weibull process. (4) Software failures can be modeled with another NHPP model. (5) HW/SW interactions can be modeled by a Markov process, considering that hardware components go to a degraded state with a rate λ_1, which causes a HW/SW failure with a rate λ_2.

The proposed model is projected to be used for hardware-software systems reliability estimation. For instance, modern embedded systems are often based on microcontrollers (i.e., microprocessors with integrated memory and peripheral interfaces) [230]. One of the applications of the proposed model is an embedded system, with the hardware block modeling the microcontroller, the software representing the firmware, and the HW/SW interactions block modeling the hardware degradation leading to software failures causing eventual system failure. The proposed generic model can be used to represent other digital systems with hardware and software components as well, by making the appropriate modifications. For example, more than three blocks could be used to represent the complexity of larger systems with more degraded states. The proposed methodology will be demonstrated with data from Los Alamos National Laboratory [224]. An example of the required data is shown in Table 4, later in the case study, section 4.4.

Since it is assumed that a system fails when pure hardware, pure software, or HW/SW interactions failure occurs, the appropriate reliability models are needed for each type of failure.

4.3.1. Hardware reliability

To estimate hardware reliability, a Weibull process [231], with an intensity function $h_{HW}(t)$ and mean value function $m_{HW}(t\ )$ is proposed to model pure hardware failure times:

$$h_{HW}(t) = \lambda\beta t^{\beta-1} \tag{42}$$

$$m_{HW}(t) = \lambda t^{\beta} \tag{43}$$

where, λ is the intensity parameter, and β is the shape parameter. Another equivalent parametrization is sometimes used considering the scale parameter θ [231], i.e., $m_{HW}(t) = \left(\frac{t}{\theta}\right)^{\beta}$ and $h_{HW}(t) = \left(\frac{\beta}{\theta}\right)\left(\frac{t}{\theta}\right)^{\beta-1}$, respectively. $\beta > 1, \beta = 1,$ or $\beta < 1$ indicates a deteriorating system, a homogeneous Poisson process, or reliability growth, respectively [231]. The reliability of hardware (R_{HW}) can be calculated as [22, 231]:

$$R_{HW}(t) = e^{-\lambda t^{\beta}} \tag{44}$$

The parameters λ and β can be estimated from failure data. Some of the reported hardware failures in system failure databases are, in reality, HW/SW interaction failures. In an effort to account for interaction failures, it has been assumed that the fraction of hardware failures, which are actually HW/SW failures, is a fixed value F [22, 228, 229]. However, this assumption does not necessarily mimic the real scenario. Even though the sensitivity analysis can be performed to analyze different scenarios of F, the actual proportion is usually unknown and most likely to be probabilistic in nature. Therefore, to calculate the hardware reliability, the parameters λ and β need to be estimated considering variability or randomness of F. To consider the variability of F, stochastic optimization of the parameters λ and β is proposed, as discussed later in this section.

In general, not all the analyzed units (systems) reported in the failure databases start operating simultaneously, and hence, there are different starting times for these different units. For instance, some units may have started operating in period 1, other units in period 2, and so on. Therefore, for the analysis purpose, hardware failure data can be divided into p groups, corresponding to p periods of time considering the starting time of the units. The periods of time can be represented by weeks, months, years, etc., depending on the availability of data. Let the starting time of the i^{th} period be s_i, for $i = 1,2, \ldots p$, and n_i be the number of units activated (started) at period i. The observed failures that occurred during a period are accounted for at the end of each time period. Let the end time of each period, i.e., the observation time be t_i, for $i = 1,2, \ldots p$. Therefore, the observation times t_i are related to the starting times as follows (Figure 14):

$$t_{i-1} = s_i \tag{45}$$

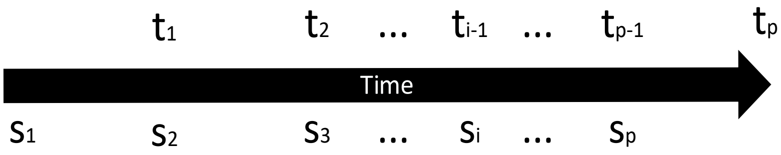


Figure 14. Starting and observation times

Let X_i be the number of HW/SW interaction failures in the period i, and e_i be the allocated number of hardware failures attributed to each period i. For each period i, the relationship between X_i and e_i is assumed to remain as [22, 23, 228]:

$$F = \frac{X_i}{e_i} \tag{46}$$

where F is the fraction of reported hardware failures that are indeed HW/SW failures. This relationship is illustrated in Figure 15,

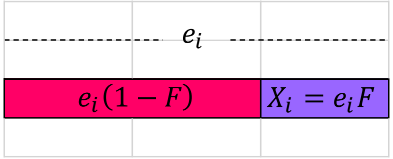


Figure 15. HW-SW interaction failures as a fraction of the failures allocated and reported as hardware failures

Assuming that each unit follows a Weibull process [231], from Equations (42) and (43), the hardware intensity function and mean value function for the r^{th} unit are given as [22]:

$$h_{HW_r}(t) = \lambda\beta\left(t - s_{i_r}\right)^{\beta-1} I\left(t \geq s_{i_r}\right) \tag{47}$$

$$m_{HW_r}(t) = \lambda\left(t - s_{i_r}\right)^{\beta} I\left(t \geq s_{i_r}\right) \tag{48}$$

where i_r is the starting period of the r^{th} unit, i.e., $i_r \in \{1,2\,, \ldots, p\}$, and $I(\cdot)$ is the indicator function.

Recalling that n_i is the number of units activated at the beginning of period i, the expected number of failures through time t_1 is given by $n_1\lambda(t_1 - s_1)^{\beta}$. Similarly, the expected number of failures through time t_2 is given by $n_1\lambda(t_2 - s_1)^\beta + n_2\lambda(t_2 - s_2)^\beta$. Generalizing, let $\mu(t_i)$ be the expected number of failures through time t_i, which can be given as:

$$\mu(t_i) = \sum_{j=1}^{i} n_j \lambda \left(t_i - s_j\right)^\beta, i = 1,2, \ldots p \qquad (49)$$

Considering hardware and software interaction, the fraction F must be removed from the total failures O_i reported as hardware failures. This fraction of failure, although reported as hardware failure in the database, treated as HW/SW interactions, will be used to fit the HW/SW interactions model in the proposed approach. With this consideration, the parameters λ and β can be estimated by minimizing the sum of squared errors (SSE):

$$SSE(\lambda, \beta) = \sum_{i=1}^{p} \left(\mu(t_i) - O_i(1 - F)\right)^2 \qquad (50)$$

where, O_i represents the cumulative number of reported hardware failures recorded at the corresponding observation times t_i.

If fixed values were assumed for *HW/SW* interaction failure fraction (*F)*, the objective value $SSE(\lambda, \beta)$ could be minimized by conventional non-linear optimization models. However, to consider the probabilistic nature of F, the proposed approach models it as a random variable. As a result of uncertainty in F, the objective function $SSE(\lambda, \beta, F)$ is also a random variable that depends not only on the decision variables $\lambda, \beta,$ but also on the random parameter F. We, therefore, propose to minimize the expected value of $SSE(\lambda, \beta, F)$. This problem falls within the category of stochastic programming [232, 233], and can be expressed as:

$$\min_{\lambda,\beta \geq 0} \mathbb{E}\left(SSE(\lambda, \beta, F)\right) \qquad (51)$$

By considering the equivalence in Equation (50), the problem can be formulated as:

$$\min_{\lambda,\beta\geq 0} \mathbb{E}\left(\sum_{i=1}^{p}\big(\mu(t_i) - O_i(1-F)\big)^2\right) \tag{52}$$

The optimal parameter values λ^* and β^* can then be used to estimate the reliability of hardware, $\hat{R}_{HW}(t)$, as given below:

$$\hat{R}_{HW}(t) = R_{HW}(t, \lambda^*, \beta^*) \tag{53}$$

4.3.2. Software reliability

Software reliability, $R_{SW}(t)$ can be modeled by predictive NHPP approaches based on software testing failure data [119-122], which have proven to fit well with the real data. For instance, they were successfully used to predict firmware (software embedded in the hardware) reliability by Hewlett-Packard and software reliability by AT&T Bell Laboratories [234]. A stochastic counting process $\{N(t), t \geq 0\}$ represents the cumulative number of errors detected by time t. $N(t)$ follows the Poisson distribution with a characteristic mean value function, $m_{SW}(t)$, which represents the expected number of accumulated software failures at time t. A review of software reliability models by Pham [120] provides a detailed discussion. Each model has a particular $m_{SW}(t, \boldsymbol{\theta})$ whose parameters set $\boldsymbol{\theta}$ can be estimated from the software failure data.

Software reliability $R_{SW}(t|T)$ is defined as the probability that a software failure does not occur in the interval $(T, T+t)$, given that the last failure occurred at testing time $T\ (T \geq 0, t > 0)$. The software component reliability is given as [120]:

$$R_{SW}(t|T) = \ exp^{-[m_{SW}(T+t) - m_{SW}(T)]} \tag{54}$$

where $m_{SW}(\cdot)$ is the software mean value function of pure software failures.

The estimated software reliability, $\hat{R}_{SW}(t|T)$ is given as:

$$\hat{R}_{SW}(t|T) = \ exp^{-[m_{SW}(T+t;\hat{\boldsymbol{\theta}}) - m_{SW}(T;\hat{\boldsymbol{\theta}})]} \tag{55}$$

where $\widehat{\boldsymbol{\theta}}$ is the vector of model parameters estimated using the maximum likelihood estimation (MLE) approach.

4.3.3. Hardware/software interactions reliability

Besides pure hardware and software component failures, the HW/SW interactions failure is also considered separately for the reliability assessment of the hardware-software system. A proportion *(F)* of the failures reported as hardware failures is considered to be related to HW/SW interaction failures. The HW/SW interactions can be modeled as a Markov process, as shown in Figure 16. In the initial state 0, the system is in a perfect working state. Over a period of time with regular usage, the hardware degrades and goes to a degraded state 1 with a transition rate λ_1. A degraded hardware state may lead to a HW/SW failure state 2 with a transition rate λ_2.

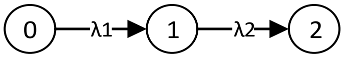


Figure 16. State transitions for hardware-software/interactions.

The equations that capture the proposed Markov process are given as:

$$Q_0'(t) = -\lambda_1 Q_0(t) \tag{56}$$

$$Q_1'(t) = \lambda_1 Q_0(t) - \lambda_2 Q_1(t) \tag{57}$$

$$Q_2'(t) = \lambda_2 Q_1(t) \tag{58}$$

where $Q_k(t)$ is the probability of the Markov process being at the k^{th} state at time t, for $k = 0,1,2$, and $Q_k'(t) = \frac{dQ_k(t)}{dt}$.

By assuming a perfect operation at the starting time, the initial conditions are:

$$Q_0(0) = 1 \tag{59}$$

$$Q_1(0) = 0 \tag{60}$$

$$Q_2(0) = 0 \tag{61}$$

The Equations (56-58), along with the initial conditions captured in Equations (59-61), thoroughly describe the process [235].

$Q_2(t)$ represents the probability of a HW/SW interaction failure. By solving the system of differential equations with the given initial conditions, $Q_2(t)$ is given as:

$$Q_2(t) = 1 - \frac{\lambda_2 \exp(-\lambda_1 t) - \lambda_1 \exp(-\lambda_2 t)}{\lambda_2 - \lambda_1} \tag{62}$$

Equation (62) is valid for $\lambda_1 \neq \lambda_2$. In the case that $\lambda_1 = \lambda_2$, let $\lambda_1 = \lambda_2 = \lambda_{HW/SW}$, $Q_2(t)$ is given as:

$$Q_2(t) = 1 - \left(\lambda_{HW/SW} t + 1\right) \exp\left(-\lambda_{HW/SW} t\right) \tag{63}$$

Therefore, since Q_2 is the probability of being at a HW/SW failure state, the reliability of HW/SW interactions $R_{HW/SW}(t)$, is given as:

$$R_{HW/SW}(t) = 1 - Q_2(t) \tag{64}$$

For the case when $\lambda_1 \neq \lambda_2$, reliability estimate is given as:

$$R_{HW/SW}(t) = \frac{\lambda_2 \exp(-\lambda_1 t) - \lambda_1 \exp(-\lambda_2 t)}{\lambda_2 - \lambda_1} \tag{65}$$

When $\lambda_1 = \lambda_2 = \lambda_{HW/SW}$, reliability of HW/SW interaction is estimated as:

$$R_{HW/SW}(t) = \left(\lambda_{HW/SW} t + 1\right) \exp\left(-\lambda_{HW/SW} t\right) \tag{66}$$

Let T_i and X_i ($X_i = Fe_i$, from equation (46)) represent the aggregate exposure time and number of HW/SW interactions failures, respectively, for each group of units that started operating at the i^{th} period. By assuming a renewal process for the HW/SW failure times for a particular unit, the approximated expected value of X_i can be obtained by [52]:

$$\mathbb{E}(X_i) = \frac{T_i}{\mu} \tag{67}$$

where μ is the mean time to HW/SW failures, which is given as:

$$\mu = \frac{1}{\lambda_1} + \frac{1}{\lambda_2} \tag{68}$$

The estimated optimal values for λ_1 and λ_2 can be obtained by using the non-linear least-squares method minimizing $SSE(\lambda_1, \lambda_2)$.

$$SSE(\lambda_1, \lambda_2) = \sum_{i=1}^{p} \left(X_i - \frac{T_i}{\frac{1}{\lambda_1} + \frac{1}{\lambda_2}} \right)^2 \tag{69}$$

A known ratio of $\frac{\lambda_2}{\lambda_1}$ would allow estimating λ_1, and λ_2 from Equation (69). However, if there is no information available on the ratio, λ_1 and λ_2 values cannot be uniquely determined. In that case, the reparameterization method can be used by letting $G = \frac{\lambda_2}{\lambda_1}$ and setting $\frac{\partial SSE}{\partial \lambda_1} = 0$. The estimated values of these parameters, $\widehat{\lambda_1}$ and $\widehat{\lambda_2}$, are given as:

$$\widehat{\lambda_1} = \left(1 + \frac{1}{G}\right) \frac{\sum_{i=1}^{p} X_i T_i}{\sum_{i=1}^{p} T_i^2} \tag{70}$$

Substituting X_i from Equation (46), the optimal estimates of $\widehat{\lambda_1}$ and $\widehat{\lambda_2}$ are given as:

$$\widehat{\lambda_1} = \left(1 + \frac{1}{G}\right) \frac{\sum_{i=1}^{p} e_i F T_i}{\sum_{i=1}^{p} T_i^2} \tag{71}$$

$$\widehat{\lambda_2} = G \widehat{\lambda_1} \tag{72}$$

Since, X_i depends on the selected value *F* (see Equation (46)), and when *F* is assumed to be a fixed value, the parameters $\widehat{\lambda_1}$ and $\widehat{\lambda_2}$ can be uniquely determined by using Equations (71) and (72). However, if *F* is assumed a random variable, the parameters $\widehat{\lambda_1}$ and $\widehat{\lambda_2}$ become random variables as well. Moreover, in this study, G is also considered as a random variable. To deal with this situation, a Monte Carlo simulation method is proposed considering a sampled distribution for *G*, which results in sampled distributions for $\widehat{\lambda_1}$, $\widehat{\lambda_2}$, and HW/SW interactions reliability, $R_{HW/SW}(t)$. The sampled median value of $R_{HW/SW}(t)$ can be used as a point estimator of the HW/SW reliability, $\hat{R}_{HW/SW}(t)$. Such estimator provides 50% confidence lower bounds. Confidence intervals and quantiles can also be obtained from a sampled distribution of $R_{HW/SW}(t)$. The HW/SW interaction reliability estimate is given as:

$$\hat{R}_{HW/SW}(t) = Sample\ median\ \left(R_{HW/SW}(t)\right) \tag{73}$$

4.3.4. System reliability

A system is conceived as the series configuration which might fail due to hardware, software, or HW/SW interaction failures. Hence, the reliability of the system depends on the reliability of the hardware (R_{HW}), software (R_{SW}), and HW/SW interaction $\left(R_{HW/SW}\right)$. Considering the series configuration and from assumptions (1) – (5), the reliability of the system, $R_{SYS}(t)$ is given as:

$$R_{SYS}(t) = R_{HW}(t)R_{SW}(t)R_{HW/SW}(t) \tag{74}$$

The system reliability point estimator is given as:

$$\hat{R}_{SYS}(t) = \hat{R}_{HW}(t)\hat{R}_{SW}(t)\hat{R}_{HW/SW}(t) \tag{75}$$

$$\hat{R}_{SYS}(t) = R_{HW}(t, \lambda^*, \beta^*)R_{SW}\left(t, \widehat{\boldsymbol{\theta}}\right)Sample\ median\ \left(R_{HW/SW}(t)\right) \tag{76}$$

In addition, sampled values of system reliability can be obtained by:

$$Sampled\ R_{SYS}(t) =\ R_{HW}(t, \lambda^*, \beta^*)R_{SW}\left(t, \widehat{\boldsymbol{\theta}}\right)Sampled\ R_{HW/SW}(t) \tag{77}$$

From the sampled values, statistical measures like mean, confidence intervals, quantiles, etc., can be calculated as well.

4.4. Case study

This section demonstrates the applicability of the proposed methodology. Failure data from a set of high-performance computing systems provided by Los Alamos National Laboratory [224] was used in this case study. Failure data of 15 systems from 1998 to 2003 were analyzed.

The selected systems (3 - 6, 8 - 14, 18 – 20, and 23) account for 96% of the nodes for which there are data available about installation dates. Each node integrates hardware and software, and therefore, was considered as a single unit (system) for the analysis. The deployment years for the nodes and the number of nodes activated are shown in Table 3. Such data were taken from the installation date of the original dataset.

Table 3. Nodes deployed per year

Year	1998	2001	2002	2003
# of nodes	5	676	2,048	1,824

Table 4. Failure data: hardware and software

Period (Year), i	Implied # of nodes	Exposure Time (System-days)	HW failures	SW failures
1 (1998)	5	1,825	6	22
2 (1999)	5	1,825	8	17
3 (2000)	5	1,825	8	15
4 (2001)	681	248,565	48	9
5 (2002)	2,729	996,085	1,486	431
6 (2003)	4,553	1,661,845	2,231	643
Total	NA	2,911,970	3,787	1,137

In the original dataset, the failures classified as hardware-related have 78 attributed causes, but we selected only the majority of causes, including CPU, Memory Dimm, Disk Drive, and Interconnect Soft Error. These account for more than 85% of the total hardware failures and some of them seem to be more related to the possibility of interaction with software failures. Similarly, for software failures, Operating System (OS), Parallel File System, Other Software, Scheduler Software, Resource Management System, Cluster File System, and Kernel software represented approximately 85 % of the total software failures and were selected to fit the software reliability model. The total number of hardware and software failures is listed in columns 4 and 5 of Table 4. Column 2 shows the implied total number of nodes working in each period calculated as the cumulative values from Table 3. The exposure time in system days is calculated by summing over all periods as shown in Table 5 (row 'Total').

Each year was considered an integer period resulting in six periods (1998 to 2003), i.e., $p = 6$. The number of deployed nodes in column 2 of Table 5 corresponds to the values are given in Table 3. The number of allocated hardware failures e_i in column 10 of Table 5 was calculated by proportionally allocating the total number of hardware failures (3,787) according to the ratios of *aggregated exposure time* to the *total aggregated exposure time* (2,911,970 days). For instance, considering the fourth period, the number of allocated failures 963 is calculated as $3{,}787 \times \frac{740{,}220}{2{,}911{,}970}$.

Table 5. Exposure time and hardware failures

Period (Year), i	# of nodes	Period and number of days						Aggregate exposure time, T_i	Allocated # of failures, e_i
		1 365	2 365	3 365	4 365	5 365	6 365		
1	5	1,825	1,825	1,825	1,825	1,825	1,825	10,950	14
2	-	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-	-
4	676	-	-	-	246,740	246,740	246,740	740,220	963
5	2,048	-	-	-	-	747,520	747,520	1,495,040	1,944
6	1,824	-	-	-	-	-	665,760	665,760	866
Total	4553	1,825	1,825	1,825	248,565	996,085	1,661,845	2,911,970	3,787

4.4.1. HW/SW interactions reliability estimation

For simplicity, G was considered to follow a uniform distribution $U(5, 10)$ as shown in Figure 17.

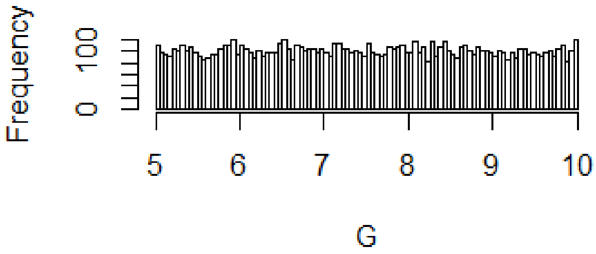


Figure 17. Simulated distribution for G in scenarios 1 to 4

Different parameters or distributions could be selected if the system under analysis would provide more information; for example, a historical ratio between the rates λ_1 and λ_2 from failure or root cause analysis data. For instance, modern hardware can provide self-monitoring alerts with timestamps reporting a degraded state. On the other hand, several distributions were considered to model *F*. Since *F* is essentially unknown, previous studies [22, 23] have assumed fixed values of *F* ranging from 0.05 and 0.25. To account for the variability, we consider *F* as a random variable that follows a distribution within the same range. As λ_1 and λ_2 will depend on the distributions of *F* and *G*, different scenarios are proposed for the analysis purpose. Four different distribution scenarios, considering uniform and truncated normal distributions, were investigated in this study:

1. $F \sim$U $(a = 0.05, b = 0.25)$
2. $F \sim$Truncated N $(\mu = 0.05, \sigma^2 = 0.03^2, a = 0.05, b = 0.25)$
3. $F \sim$Truncated N $(\mu = 0.15, \sigma^2 = 0.03^2, a = 0.05, b = 0.25)$
4. $F \sim$Truncated N $(\mu = 0.25, \sigma^2 = 0.03^2, a = 0.05, b = 0.25)$

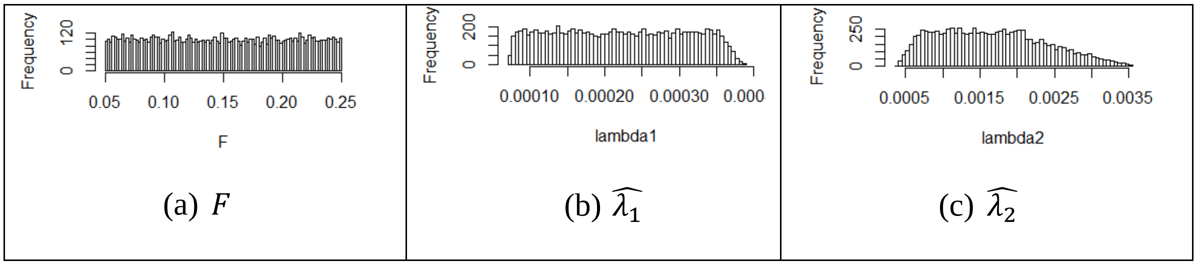
The distributions for scenario 1 and scenario 3 cover smoothly and symmetrically the entire domain (0.05, 0.25) for the random variable *F*. The distributions for scenarios two and four aim to model extreme cases of distribution of *F* with modes at 0.05 and 0.25, respectively.

The truncated normal distribution is used to preserve a smooth distribution but limiting F to the predefined domain (0.05, 0.25).

Rather than deriving the exact mathematical distributions for $\widehat{\lambda_1}$ and $\widehat{\lambda_2}$, which will result in different expressions depending on the assumed distributions for G and F, a practical simulation approach was used to get quick results. R code was developed for this purpose. The standard function *runif* of R was used to generate samples from uniform distributions, whereas the package ‘truncnorm’ [236] was used to generate samples from truncated normal distributions. ‘matrixStats’ package [237] was used to calculate quantiles of the sampled distributions. Reliability prediction was calculated within approximately 1.5 seconds for a 1,095-day horizon, which demonstrated the feasibility and convenience of the proposed approach. Figs. 18 - 21 show the resultant sampled distributions of $\widehat{\lambda_1}$ and $\widehat{\lambda_2}$ obtained by simulating 10,000 values of *F* and *G*.

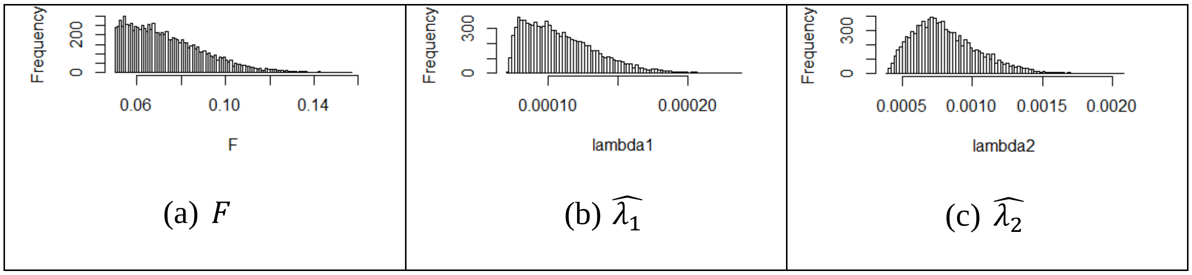
It is worth mentioning that $\widehat{\lambda_1}$ and $\widehat{\lambda_2}$ do not necessarily follow distributions similar to the distribution of *F*. For instance, the resultant distributions of $\widehat{\lambda_1}$ and $\widehat{\lambda_2}$ in Figure 18 b and c, respectively, look trapezoidal, whereas *F* was assumed uniform. A similar result is obtained for $\widehat{\lambda_2}$, which looks trapezoidal in Figure 21 c when it is assumed that *F* follows Truncated N $(\mu = 0.25, \sigma^2 = 0.03^2, a = 0.05, b = 0.25)$.

The HW/SW reliability $R_{HW/SW}(t)$ at any given point of time is given by Equation (65), and since $\widehat{\lambda_1}$ and $\widehat{\lambda_2}$ are random variables, $R_{HW/SW}(t)$ is also a random variable. To visualize this situation, consider an example at $t = 1$ with F ~Truncated N $(\mu = 0.15, \sigma^2 = 0.03^2, a = 0.05, b = 0.25)$. The simulated distribution of $R_{HW/SW}(1)$ values are shown in Figure 22.



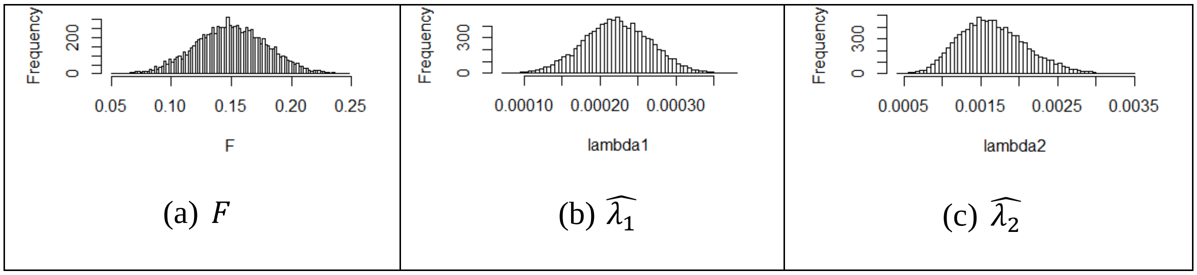
(a) F (b) $\widehat{\lambda_1}$ (c) $\widehat{\lambda_2}$

Figure 18. Simulated distributions for scenario 1



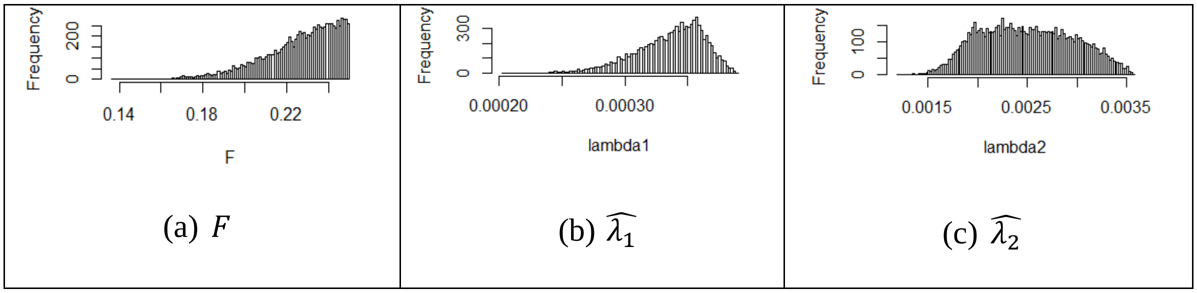
(a) F (b) $\widehat{\lambda_1}$ (c) $\widehat{\lambda_2}$

Figure 19. Simulated distributions for scenario 2



(a) F (b) $\widehat{\lambda_1}$ (c) $\widehat{\lambda_2}$

Figure 20. Simulated distributions for scenario 3



(a) F (b) $\widehat{\lambda_1}$ (c) $\widehat{\lambda_2}$

Figure 21. Simulated distributions for scenario 4

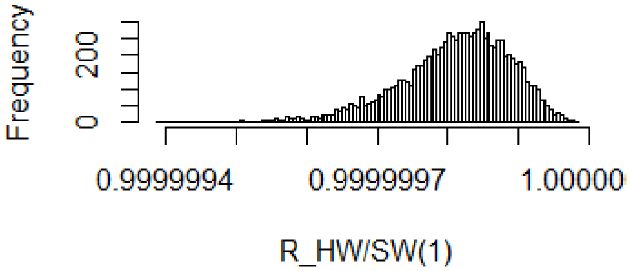


Figure 22. Simulated distribution for $R_{HW/SW}(1)$ assuming F ~Truncated N $(\mu = 0.15, \sigma^2 = 0.03^2, a = 0.05, b = 0.25)$

Summary statistics for sampled $R_{HW/SW}(t)$ considering different distributions for *F* are shown in Tables 6-9. The widest confidence intervals (Table 6) are obtained when *F* is assumed to follow a uniform distribution, which can be expected because the distribution of *F* is uniformly spread over the entire range (0.05, 0.25). Additionally, the distribution with the mode at 0.25 generates a lower $R_{HW/SW}(t)$ reliability than the distribution with mode at 0.05. This occurs because a mode at 0.25 indicates higher HW/SW interaction than having a mode at 0.05.

Table 6. Summary statistics of HW/SW interactions reliability for scenario 1

t (Days)	1	548	1,095
95% C.I.	0.9999995 1.0000000	0.9087141 0.9935149	0.7665665 0.9768527
Median	0.9999998	0.9609935	0.8824369
Mean	0.9999998	0.957793	0.8792081

Table 7. Summary statistics of HW/SW interactions reliability for scenario 2

t (Days)	1	548	1,095
95% C.I.	0.9999999 1.000000	0.9736864 0.9948450	0.9176090 0.9813172
Median	1.0000000	0.989682	0.9644654
Mean	1.0000000	0.9882022	0.9603000

Table 8. Summary statistics of HW/SW interactions reliability for scenario 3

t (Days)	1	548	1,095
95% C.I.	0.9999996 0.9999999	0.9297811 0.9839945	0.8104245 0.9464829
Median	0.9999998	0.9606466	0.8818387
Mean	0.9999998	0.9596984	0.8809555

Table 9. Summary statistics of HW/SW interactions reliability for scenario 4

t (Days)	1	548	1,095
95% C.I.	0.9999994 0.9999998	0.9025302 0.9474176	0.7562508 0.8461515
Median	0.9999996	0.9216735	0.7890623
Mean	0.9999996	0.9224262	0.7923520

The median value of the sampled $R_{HW/SW}(t)$ can be used as a point estimation of the HW/SW reliability, $\hat{R}_{HW/SW}(t)$. Such estimators provide 50% confidence lower bounds.

4.4.2. Hardware reliability estimation

The data in the form of starting times s_i, nodes activated n_i, observation times t_i, and cumulative failures O_i are shown in Table 10. n_i comes from Table 3. O_i represents the cumulative hardware failures given in column 4 of Table 4. Starting times s_i and observation times t_i are the cumulative times corresponding to the 6 periods, each period having 365 days. The relationship between s_i and t_i is given by Equation (45).

$\hat{R}_{HW}(t) = R_{HW}(t, \lambda^*, \beta^*)$.

Table 10. Cumulative failures, nodes, starting and observation times

Start Time s_i (Days)	Nodes Activated n_i	Observation Time t_i (Days)	Cumulative Failures O_i
0	5	365	6
365	0	730	14
730	0	1,095	22
1,095	676	1,460	70
1,460	2,048	1,825	1,556
1,825	1,824	2,190	3,787

The model was formulated and solved using GAMS solver with EMP framework [222] to find the optimal values of λ and β that minimize the expected value of $SSE(\lambda, \beta)$ in Equation (49) using the data from Table 10. GAMS solves the problem by using Monte Carlo sampling method [222]. The Monte Carlo sampling method generates a finite number of scenarios to approximate the continuous distributions assumed for F, and the problem is converted to a problem with a finite discrete distribution. The results are summarized in Table 11.

Table 11. Stochastic optimization of parameters λ, β

Scenario	λ^*	β^*	$E(SSE)$
1	0.00020228	1.25874892	36457.29
2	0.00022106	1.25874892	43346.10
3	0.00020269	1.25874892	36472.11
4	0.00018465	1.25874892	30251.18

It is worth to notice that the parameter β is not sensitive to the distribution of F, and it is greater than one, which indicates the wearing out or degradation behavior of the hardware. The optimal parameters $\lambda^*$and β^* are used in Equation (44) to estimate the reliability of hardware, $\hat{R}_{HW}(t) = R_{HW}(t, \lambda^*, \beta^*)$.

4.4.3. Software reliability estimation

An inflection S-shaped model [223] was chosen considering the S-shape of the available data and because it requires only three parameters. This model has been reported to generate the best fitting based on the SSE for an S-shaped dataset [120]. Figure 23 shows the actual number of software failures, $m_{SW}(t)$ and the MLE estimated number of failures, $\widehat{m}_{SW}(t)$.

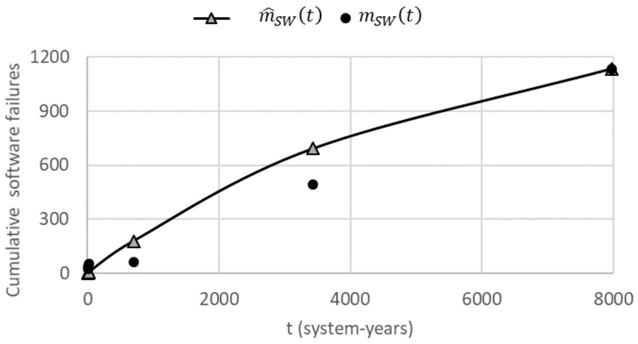


Figure 23. Cumulative software failures: actual software failures from data, $m_{SW}(t)$ and estimated software failures with $\widehat{m}_{SW}(t)$

The S-shaped model is defined by Equations (78) and (79).

$$D(t) = \frac{d}{1 + \gamma e^{-dt}} \tag{78}$$

where $D(t)$, a logistic function, represents the failure detection rate per fault, d is the failure-detection rate, and γ is the inflection factor. The mean value function is given by:

$$m(t) = \frac{c}{1 + \gamma e^{-dt}}(1 - e^{-dt}) \tag{79}$$

MLE parameters can be estimated by solving a system of equations [120].

To fit the model, the testing times and software failures from the third and fifth columns of Table 4 were used. For computational convenience, the testing times in system-days were converted to system-years, as shown in Table 12 and Figure 23. The MLE parameter estimates are: $\hat{c} = 1468.8670801$, $\hat{d} = 0.00018739$, $\hat{\gamma} = 0.00970647$. Finally, the estimated software reliability $\hat{R}_{SW}(t/T)$, where t and T are in days is, is given as:

$$\hat{R}_{SW}(t/T) = \ exp^{-\left[\hat{m}\left(\frac{T+t}{365}\right) - \hat{m}\left(\frac{T}{365}\right)\right]} \tag{80}$$

where $\hat{m}(t) = m\big(t, \hat{c}, \hat{d}, \hat{\gamma}\big)$.

Table 12. Software failure data

Period (Year), i	Exposure Time (System-days)	Exposure Time (System- years)	SW failures	Cumulative SW failures
1	1,825	5	22	22
2	3,650	10	17	39
5	5,475	15	15	54
4	254,040	696	9	63
5	1,250,125	3,425	431	494
6	2,911,970	7,978	643	1,137

4.4.4. System reliability estimation

The point estimate of system reliability is the product of hardware, software, and HW/SW interactions reliability:

$$\hat{R}_{SYS}(t) = \hat{R}_{HW}(t)\hat{R}_{SW}(t)\hat{R}_{HW/SW}(t) \tag{81}$$

$$\hat{R}_{SYS}(t) = R_{HW}(t, \lambda^*, \beta^*)R_{SW}\big(t, \hat{c}, \hat{d}, \hat{\gamma}\big)Median\left(Sampled\ R_{\frac{HW}{SW}}(t)\right) \tag{82}$$

Sampled values of system reliability can be obtained by:

$$Sampled\ R_{SYS}(t) = R_{HW}(t, \lambda^*, \beta^*)R_{SW}\big(t, \hat{c}, \hat{d}, \hat{\gamma}\big)Sampled\ R_{HW/SW}(t) \tag{83}$$

From sampled values, statistical measures like confidence intervals, quantiles, and mean can be calculated as well.

The system reliability estimates based on the proposed approach are plotted in Figs. 24 - 27, for scenarios 1 – 4, respectively. For each scenario, the system reliability median (Rsys

(Median)) is plotted along with a 95% confidence interval shaped by the lower bound (Rsys LB) and the upper bound (Rsys UB). It is worth noting that among the four tested distributions, the scenario with $F \sim U(a = 0.05, b = 0.25)$ provides the widest 95% confidence interval which could be explained because of the “large” spread of F under uniform distribution assumption.

The proposed approach is compared with the reliability obtained under no HW/SW interactions assumption, denoted by Rsys_indep. To estimate Rsys_indep, all hardware failure data are used without deducting or considering HW/SW interactions failure data. Therefore, under this assumption of independent hardware and software failures, the reliability of the system is just the product of hardware reliability and software reliability. Rsys_indep is included in Figures 24 - 27 for comparison. For better visualization of the difference between Rsys and Rsys_indep, zoomed images are placed on the top right corner of Figures 24 - 27.

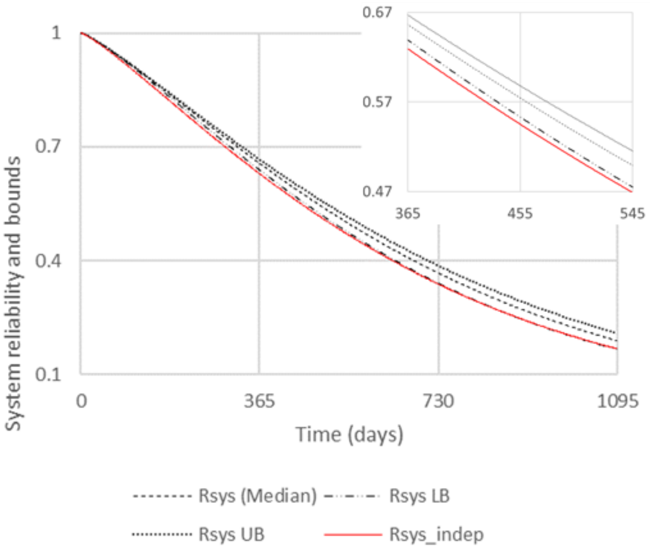


Figure 24. System reliability scenario 1

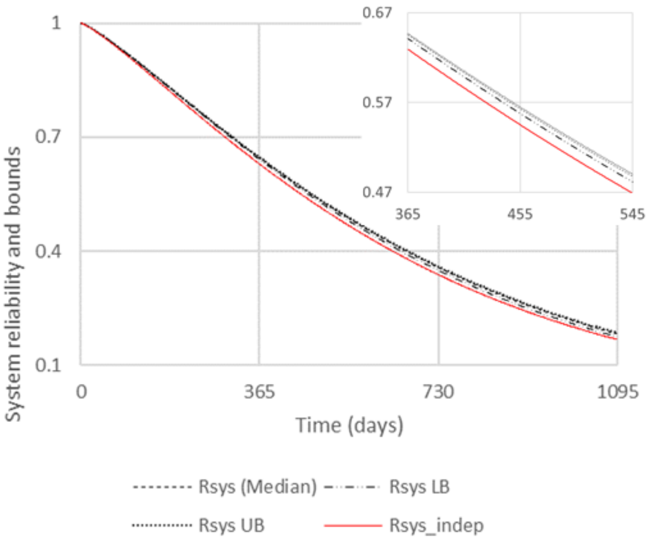


Figure 25. System reliability scenario 2

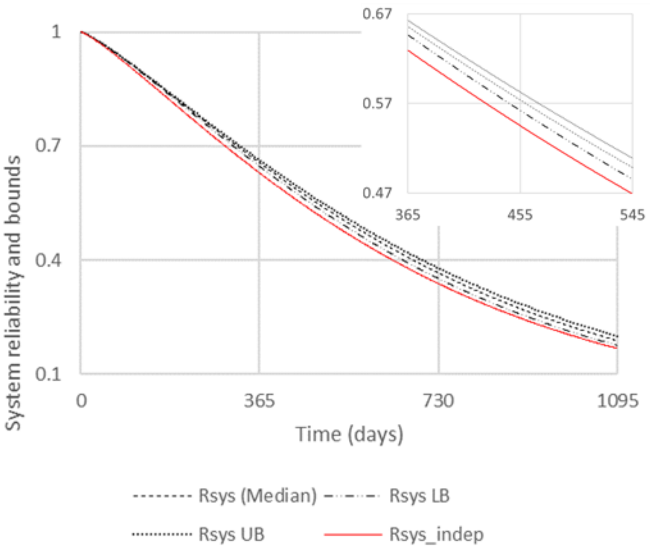


Figure 26. System reliability scenario 3

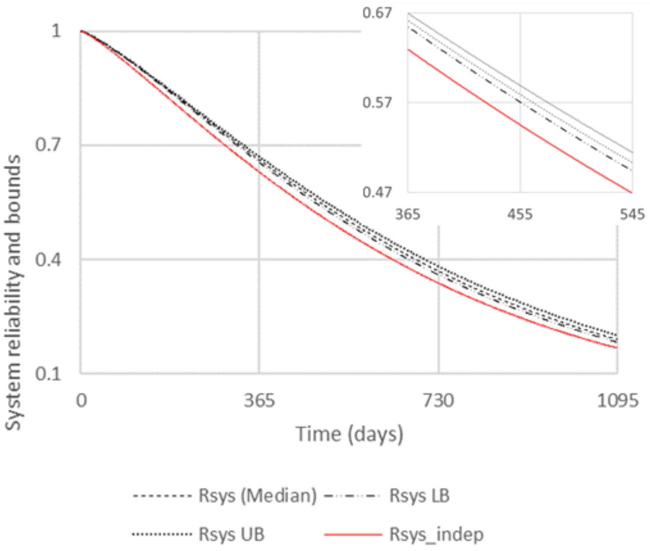


Figure 27. System reliability scenario 4

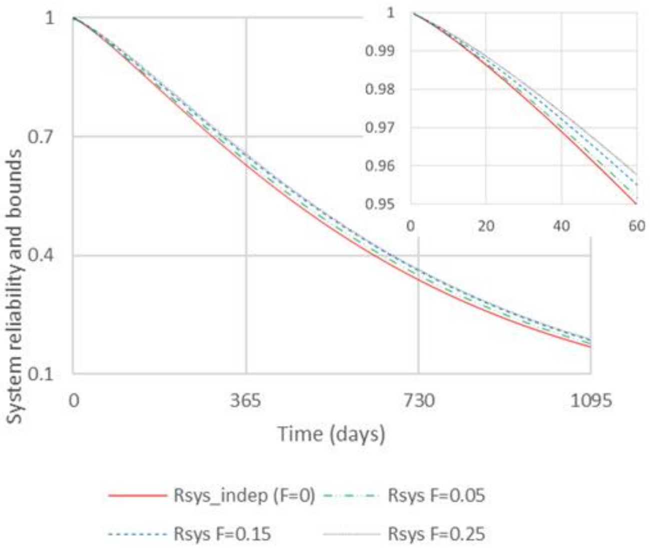


Figure 28. System reliability with fixed values $F = 0.25, 0.15, 0.05, and\ 0, G = 10$

It can be seen that for all the assumed distributions for F, the median reliability (Rsys (Median)) is consistently above the Rsys_indep as shown in Figs. 24 - 27. Interestingly, even the 97.5% lower bound (0.025 quantile) of Rsys is always greater than Rsys_indep, which means that Rsys_indep underestimates the “true” reliability. Moreover, From Figure 25 (distribution of F with the mode at 0.05) to Figure 27 (distribution of F with the mode at 0.25), it is observed that for smaller mode values of *F* the system reliability (Rsys) confidence interval is closer to Rsys_indep, which seems plausible because smaller *F* values are "closer" to no HW/SW interaction assumption. This underestimation trend of Rsys_indep is revealed even when fixed values of *F* and *G* are considered, a common assumption in existing literature [22, 228, 229]. For instance, Figure 28. shows the system reliability calculated with fixed values $F = 0.25, 0.15, 0.05, \text{and } 0, G = 10$. Particularly, if *F* has a fixed value of 0, it is equivalent to no HW/SW interactions, which is also included in Figure 28. In all the combinations, the reliability obtained by assuming independence Rsys_indep (*F*=0) underestimated the "true" reliability Rsys calculated by considering *HW/SW* interaction with fixed values (*F*>0, *G*>0*)*. The underestimation of Rsys_indep is therefore consistent for both types of assumptions for *F* and *G*, i.e., probabilistic and fixed values.

It is worth noting that we considered a three-component series configuration for calculating Rsys. In contrast, a two-component series configuration is assumed for Rsys_indep. Interestingly, Rsys estimate turns out to be greater than Rsys_indep. To investigate this outcome, a few numerical examples are considered.

Table 13 shows extreme cases of *F* and *G* combinations. These cases illustrate and corroborate the underestimation of system reliability when it is assumed that there are no HW/SW interaction failures, i.e., assuming *F* = 0. It can be observed from Table XI that the lowest system reliability estimates (Rsys) occur for the combination of *F* = 0.25 with *G* = 10. For instance, at time t=1 day, Rsys(1) = 0.9996503 > Rsys_indep(1) = 0.9995912. Similarly, at time t=1095 days, Rsys(1095) = 0.1873180 > Rsys_indep(1095) = 0.1676949. The reasoning of underestimation of Rsys_indep can be explained by comparing the differences in the individual components of system reliability, i.e., R_{HW}, R_{SW}, and $R_{HW/SW}$. Consider the case at time t=1 day, under HW/SW interaction assumption, Rsys(1)=0.9996503 is the product of $R_{HW}(1) = 0.9998209$, $R_{SW}(1) = 0.9998$, and $R_{HW/SW}(1) = 0.9999994$. On the other hand, under the independence assumption, Rsys_indep(1)=0.9995912 is the product of $R_{HW}(1) = 0.9997612$ and $R_{SW}(1) = 0.9998$. However, for calculating Rsys_indep(1), $R_{HW/SW}(1)$ could be considered as equal to 1 because no HW/SW interactions are assumed. The first change occurs in R_{HW}, which is higher if HW/SW interactions are assumed because a fraction of HW failures are deducted from the "hardware" failures category and considered as HW/SW failures. This change resulted in a relative improvement of R_{HW} (difference between 0.9998209 and 0.9997612) by $+5.971 \times 10^{-5}$. Another change occurs in $R_{HW/SW}$ because of the HW/SW interactions assumption. This causes a relative reduction in $R_{HW/SW}$ (difference between 1 and 0.9999994) by 3.802×10^{-7} in $R_{HW/SW}$. There is no change in R_{SW} because software failure data remains unchanged. A closer look at changes in reliability estimates of individual components (HW, SW, or HW/SW) shows a significant increase in R_{HW} as compared to the drop-in $R_{HW/SW}$. This results in higher reliability estimate Rsys as compared to Rsys_indep when HW/SW interactions are considered.

The proposed model is also compared with the existing approaches that assume fixed values for *F* and *G* [22, 228, 229]. Under the assumption of fixed values of *F* and *G*, these approaches provide point estimates for reliability, whereas our proposed approach provides a sampled distribution of reliability estimates and, therefore, is able to generate a confidence interval. Indeed, assuming fixed values for *F* and *G* would be a particular case of our proposed model by assuming degenerate distributions [238] for *F* and *G*. Since two scenarios, scenario 1 ($F \sim U(a = 0.05, b = 0.25$), and scenario 3 $\left(F \sim \text{Truncated N } (\mu = 0.15, \sigma^2 = 0.03^2,\ a = 0.05,\ b = 0.25)\right)$, have a mean value for *F* = 0.15 and for *G* = 7.5, we consider these two scenarios to compare the proposed approach with existing methods treating fixed values *F* = 0.15 and *G* = 7.5. Figures 29 and 30 show the confidence intervals generated with the proposed approach enclosing the reliability estimated obtained with fixed *F* and *G* values. Moreover, the median curve generated considering HW/SW interactions follows closely the point estimator curve obtained with fixed values of *F* and *G*. Realizing the uncertainty or randomness in *F* and *G* values, the proposed probabilistic proposed provides a possible range of reliability estimates and hence, helps quantify uncertainty in reliability estimates. Manufacturers find this very helpful for warranty planning and spare-parts inventory management, knowing the uncertainty in estimates and preparing worst-case scenarios. On the other hand, the consideration of fixed values only provides point estimates of reliability prediction, which is not of much use. However, the reliability band can be used for better tuning of the maintenance and/or warranty policies as more information is available to manufacturers/customers, e.g., a pessimistic scenario is given by the lower bound reliability.

Table 13. System reliability estimation under extreme cases for *F* and *G*. Comparison between Rsys and Rsys_indep (*F*=0)

t(days)	F	G	λ_1	λ_2	λ	β	$\hat{R}_{HW}(t)$	$\hat{R}_{SW}(t)$	$\hat{R}_{HW/SW}(t)$	$\hat{R}_{SYS}(t)$
1	0.25	5	0.00039	0.00195	0.00018	1.26	0.9998209	0.9998	0.9999996	0.9996505
1	0.25	10	0.00036	0.00358	0.00018	1.26	0.9998209	0.9998	0.9999994	0.9996503
1	0.00	NA	NA	NA	0.00024	1.26	0.9997612	0.9998	NA	0.9995912
1,095	0.25	5	0.00039	0.00195	0.00018	1.26	0.3013019	0.8302	0.7858804	0.1965769
1,095	0.25	10	0.00036	0.00358	0.00018	1.26	0.3013019	0.8302	0.7488650	0.1873180
1,095	0.00	NA	NA	NA	0.00024	1.26	0.2019973	0.8302	NA	0.1676949

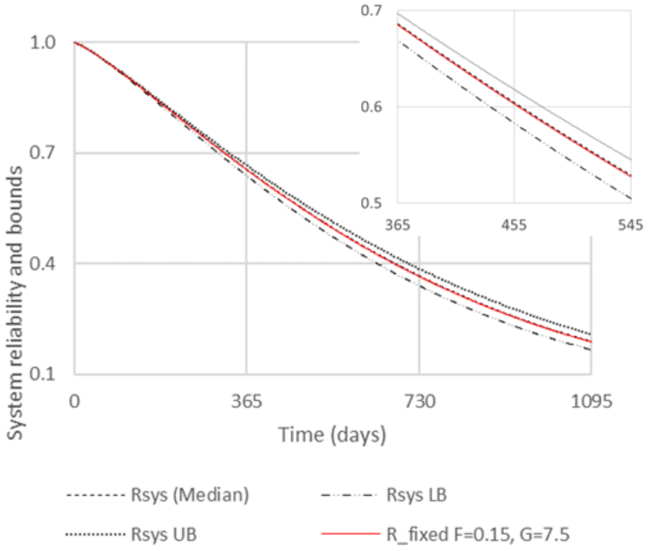


Figure 29. System reliability: calculated with the proposed method under scenario 1 $\left(F \sim U(a = 0.05, b = 0.25)\right)$ and with fixed values $(F = 0.15, G = 7.5)$

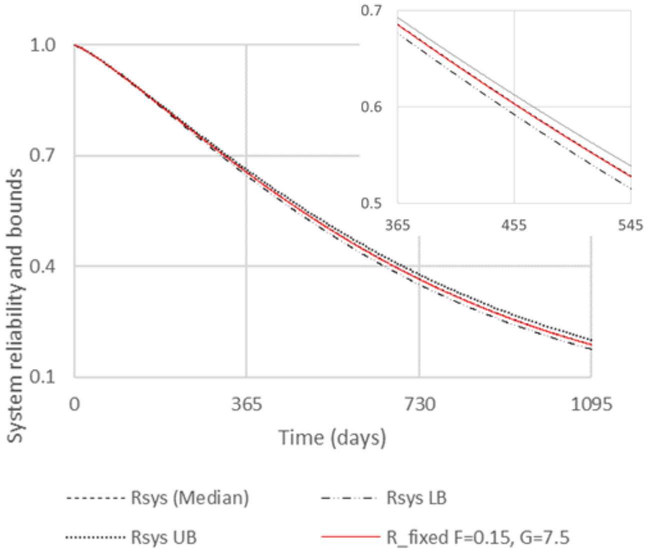


Figure 30. System reliability: calculated with the proposed method under scenario 3 and with fixed values $(F = 0.15, G = 7.5)$

4.5. Discussion

A combined hardware-software reliability model is considered in this chapter to include the effect of three components, i.e., hardware, software, and HW/SW interactions, in the reliability assessment of a system. Different from previous works, this study incorporates uncertainty in the proportion of failures attributed to hardware that are in reality caused by HW/SW interactions. The proposed study treats the fraction of hardware failures *F*, representing HW/SW interaction failure as a random variable, and a Markov process is applied to model the HW/SW interaction failures. A Weibull process and an S-shaped NHPP are employed to model hardware and software failures, respectively. The stochastic optimization approach is used to estimate the parameters of the Weibull process and hardware reliability. Additionally, since failure databases do not usually provide information on transition rates of HW/SW interactions, a ratio (*G*) of transition rates is considered. The proposed approach considers this transition rate ratio value a random variable to capture the uncertainty. Monte Carlo simulation is applied to generate samples for distributions assumed for *F* and *G*.

Real data were used to demonstrate the applicability of the proposed approach. Four scenarios were investigated by assuming different distributions for the random variables *F* and *G*. Reliability estimates for a 1,095-day horizon were provided for each scenario with an R script in approximately 1.5 seconds running on a laptop, which demonstrates the effectiveness and convenience of the approach. While analyzing all scenarios, it was found that the generated confidence intervals properly enclosed the point estimates obtained by assuming fixed values of *F* and *G*. The reliability band not only quantifies uncertainty in reliability estimates but also provides additional information that can assist manufacturers in defining better warranty and maintenance policies. It helps manufacturers to prepare for worst-case scenarios considering the lower bound of reliability if needed. The proposed methodology could be applied to embedded systems, computers, data centers, CPSs, or more general systems that integrate hardware and software.

It is also observed that reliability assessment models with no HW/SW interaction failures, assuming both hardware and software failures are independent, underestimate the system reliability. This is true for all the scenarios considered in the study, including fixed values of *F* and *G*. This finding suggests that even under higher uncertainty conditions of HW/SW interactions, it is possible to provide more reasonable reliability estimates.

CHAPTER 5. ALL-TERMINAL NETWORK RELIABILITY ESTIMATION WITH DEEP LEARNING APPROACHES[3]

5.1. Abstract

Usually, networks represent complex critical CPSs such as computer networks, piping systems, or power supply systems [40, 239, 240]. A network can be defined as a set of items (nodes or vertices) connected by edges or links [128]. Estimating the all-terminal network reliability by using artificial neural networks (ANNs) has emerged as a promissory alternative to classical exact NP-hard algorithms. Approaches based on traditional ANNs have usually considered the network reliability upper bound as part of the inputs, which implies additional time-consuming calculations during both training and testing phases. This chapter proposes the use of Convolutional Neural Networks (CNNs), without the reliability upper-bound as an input, to address the all-terminal network reliability estimation problem. The present study introduces a multidimensional matrix format to embed the topological and link reliability information of networks. The unique contribution of this work is the method to capture the topology of a network in terms of its adjacency matrix, link reliability, and topological attributes providing a novel use of CNN beyond image classification. Since CNNs have been successful for image

[3] The present chapter is based on the following papers:

1. A. Davila-Frias and O. P. Yadav, "All-terminal network reliability estimation using convolutional neural networks,". Published in Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability, p. 1748006X20969465, 2020.
   Contribution of Alex Davila Frias: developing the mathematical models, analysis of the case study, discussion of the results, and drafting the paper.
   Contribution of Om Yadav: verification of the results and proofreading the draft paper.

2. A. Davila-Frias, S. Salem, and O. P. Yadav, "Deep Neural Networks (DNNs) For All-Terminal Network Reliability Estimation," 2021. In 2021 Annual Reliability and Maintainability Symposium (RAMS). Presented.
   Contribution of Alex Davila Frias: developing the mathematical models, analysis of the case study, discussion of the results, and drafting the paper.
   Contribution of Saeed Salem and Om Yadav: verification of the results and proofreading the draft paper.

classification, appropriate modifications are needed and introduced to use them in the estimation of network reliability. A regression output layer is proposed, preceded by a sigmoid layer to achieve predictions within the range of reliability characteristic, a feature that some previous ANN-based works lack. Several training parameters together with a filter multiplier (CNN architecture parameter) were investigated. The actual values and the ones predicted with the best trained CNN were compared in the light of RMSE (0.04406) and p-value (0.3) showing non-significant difference. This study provides evidence supporting the hypothesis that the network reliability can be estimated by CNNs from its topology and link reliability information, embedded as an image-like multidimensional matrix. Another important result of the proposed approach is the significant reduction in computational time. An average of 1.18 ms/network was achieved by the CNN, whereas backtracking exact algorithm took around 500 s/network. This CNN approach is based on our paper published in the *Journal of Risk and Reliability* [75] and is presented in section 5.2.

In addition, in section 5.3 we present a DNN approach able to estimate the reliability of varying size networks. This DNN approach is based on our paper accepted for *RAMS 2021* conference [76]. To use DNNs for all-terminal network reliability estimation, an appropriate architecture of DNN needs to be developed. Different architectures are investigated by exploring parameters such as the number of hidden layers and the dropping probability parameter of a dropout layer to prevent overfitting. Hyperbolic tangent activation function limits the output to the range [0,1]. In addition, the network topology information needs to be preprocessed for the DNN to be able to process it and predict the network reliability. Graphs are used to represent the networks. Furthermore, to turn graphs into a computationally digestible format, advanced graph embedding methods (GEM) are employed. Different embedding methods and architectures are investigated together by training them with the network reliability as target. The best DNN proposed here, based on the RMSE (0.01), outperforms a previous traditional ANN approach. There is also a significant computation time reduction achieved by using the proposed DNN, which does not require the reliability upper bound as an additional input as employed in previous studies based on ANN.

5.2. Network reliability estimation with CNN [75]

5.2.1. Introduction

Usually, networks represent critical systems such as computer networks, piping systems, or power supply systems [40, 239, 240]. A network can be defined as a set of items (nodes or vertices) connected by edges or links [128]. Graphical models allow to visualize the interdependencies of the components in a system. Nodes characterize components and junctions of the system, and edges represent the connections. For example, busbars in power systems or switches in telecommunication systems are modeled by nodes, whereas edges characterize power lines in power systems and optical fibers in telecommunication systems. Such graphical models are commonly based on graph theory (GT), where a graph $G\ (N, L)$ denotes the graph G composed by the set N of nodes and the set L of links or edges [36-38].

A network is modeled by a probabilistic graph $G = (N, L, p_L)$, where N is the set of nodes, L is the set of links, and p_L is the link reliability.

At any time, only some links of G might be operational. A state of G is a sub-graph (N, L'), where L' is the set of operational links, $L' \subseteq$ L. The all-terminal network reliability of state $L' \subseteq$ L is [69]:

$$R(G) = \sum_{\Omega} \left[\prod_{j \in L'} p_L \right] \left[\prod_{j \in (L \setminus L')} (1 - p_L) \right] \tag{84}$$

Due to the exponential growth of number of states with the size of networks, the all-terminal reliability calculation is an NP-hard problem [241]. As a part of the machine learning techniques, artificial neural networks (ANNs) have emerged as a promissory tool to estimate network reliability. ANNs have been usually trained with the network topology and edge reliability as inputs and with the target network reliability as desired output [42, 45, 131]. For example, Srivaree- Ratana et al. [42] utilized an ANN to predict the all-terminal network reliability; with the network architecture, the link reliability, and the network reliability upper bound (an approximation of network reliability which is not lower than the exact value [40, 46, 242]) as inputs, and the exact network reliability as the target. More recently, Altiparmak et al. [45] proposed an ANN model to predict the all-terminal network reliability, which takes the node degree and other connectivity metrics and the upper bound network reliability as inputs to predict the network reliability.

Approaches based in traditional ANNs have usually considered the reliability upper bound as part of the inputs, which implies additional time-consuming calculations during both training and testing phases. This study proposes the use of Convolutional Neural Networks (CNNs), without the reliability upper-bound as an input, to address the all-terminal network reliability estimation problem. The approach introduces a multidimensional matrix format to embed the topological and link reliability information of networks. Since CNNs have been successful for image classification [243, 244], appropriate modifications are needed and introduced in the present study, to use them in the estimation of network reliability. A regression output layer is proposed, preceded by a sigmoid layer to achieve predictions within the physical range of reliability, a feature that some previous ANN-based works lack. Several training parameters together with a filter multiplier (CNN architecture parameter) will be investigated.

The actual values and the ones predicted with the best CNN will be compared in the light of RMSE and p-value.

5.2.2. The proposed CNN method

This section proposes a CNN approach to estimate the all-terminal reliability of networks with a given number of nodes and a given set of possible links reliability values. The proposed model makes the following assumptions for the networks:

1. The nodes are perfectly reliable.
2. The links failure probabilities are independent [245, 246].
3. The link failure probabilities are equal.
4. Each link is bi-directional.

It is hypothesized in this study that the all-terminal network reliability can be estimated from its topology and link reliability information embedded as an image-like matrix which can be processed by CNNs. In this section, the proposed approach to capture the network topology as well as topological attributes of networks is presented. The information is stacked in a multidimensional matrix to meet the "image" format that a CNN can process.

The topology of a network can be captured by its adjacency matrix. Given a graph $G = (N, L)$, with $|N| = n$ nodes, it can be represented by its adjacency matrix, i.e., a $n \times n$ symmetric matrix $\boldsymbol{A}$ defined as [247]:

$$\boldsymbol{A}(i,j) = \begin{cases} 1 & if\ n_i\ id\ adjacent\ to\ n_j \\ 0 & otherwise \end{cases} \tag{85}$$

As an example, consider the sample network represented in Figure 31. Its adjacency matrix is given in equation (86), and the resultant image of the adjacency matrix is shown in Figure 32, where a "1" corresponds to a white pixel and a "0" corresponds to a black pixel.

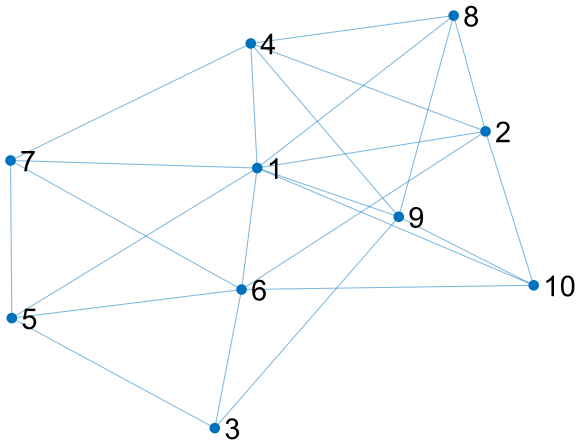


Figure 31. Sample network with 10 nodes

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \qquad (86)$$

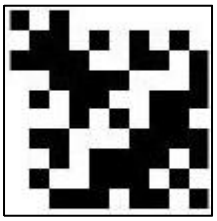


Figure 32. Image of the adjacency matrix corresponding to the sample network

To provide more information to the CNN, topological attributes such as the nodes degree and the clustering coefficient are proposed to be part of the input. The node degree [41, 248] is believed to provide network reliability information. The node degree of a node n_i is the number of its neighbors. A more general definition of the node degree, for weighted graphs, is given as [247]:

$$d_i = \sum_j \boldsymbol{A}(i,j) \tag{87}$$

The clustering coefficient is considered a measure of the disconnectedness of a graph [249] and its cliquishness [250]. Let $G_i = (N_i, L_i)$ be a subgraph induced by the neighbors of the node n_i. Let $|N_i| = n_i$ be the number of neighbors of n_i, and $|L_i| = m_i$ be the number of edges among the neighbors of n_i. The clustering coefficient of n_i is defined as [247]:

$$C(n_i) = \frac{2m_i}{n_i(n_i - 1)} \tag{88}$$

In addition, the link reliability has a direct impact in the overall network reliability. Therefore, it is considered as part of the input as well.

The proposed input format consists of layers of two-dimensional matrices concatenated along the third dimension. In the first layer, there is the adjacency matrix, like the one in equation (86). The diagonal of zeros of the adjacency matrix is replaced by the node degree, calculated using equation (87). The second layer is a diagonal matrix with the clustering coefficient calculated with equation (88). The third layer is a diagonal matrix with the link reliability. The resultant input format for each network is therefore a $n \times n \times 3$ matrix, i.e., a three-dimensional matrix. The proposed matrix might be enriched as needed by adding more layers with more information in matrix format. In this chapter we consider that the t^{th} network in the set of networks can be represented by the matrix $\boldsymbol{X_t}$, where $\boldsymbol{X_t}$ is formed by stacking three matrices: $\boldsymbol{M_{t1}}, \boldsymbol{M_{t2}}, \boldsymbol{M_{t3}}$.

$$\boldsymbol{M_{t1}} = \boldsymbol{A_t} + diag(d_{t1}, d_{t2}, d_{t3}, \dots, d_{tn}\ ) \tag{89}$$

where $\boldsymbol{A_t}$ is the adjacency matrix of the t^{th} network in the dataset, d_{ti} is the node degree of the i^{th} node of the t^{th} network, and $diag(\cdot)$ is the diagonal matrix of a vector.

$$\boldsymbol{M_{t2}} = diag\big(C(n_{t1}), C(n_{t2}), C(n_{t3}), \dots, C(n_{tn})\big) \tag{90}$$

where $C(n_{ti})$ is the clustering coefficient of the i^{th} node of the t^{th} network.

$$\boldsymbol{M_{t3}} = p_{tL} diag(\mathbf{1_n}) \tag{91}$$

where, p_{tL} is the link reliability of the t^{th} network and $\mathbf{1_n}$ is an *n*-sized row vector of ones.

A graphical representation of a matrix $\boldsymbol{X_t}$ for a network with 10 nodes is shown in Figure 33.

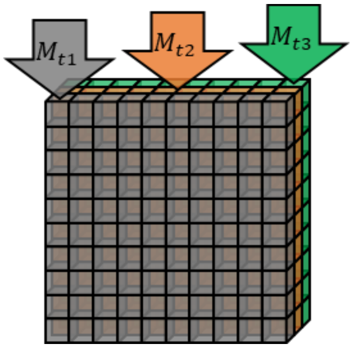


Figure 33. Graphical representation of the matrix $\boldsymbol{X_t}$, formed by stacking three matrices: $\boldsymbol{M_{t1}}, \boldsymbol{M_{t2}}, \boldsymbol{M_{t3}}$

To illustrate this format, let us assume that the sample network shown in Figure 31, whose adjacency matrix is given in equation (86), is the first network of the set. Then, the first network can be represented by $\boldsymbol{X_1}$, where $\boldsymbol{X_1}$ is formed by stacking three matrices $\boldsymbol{M_{11}}, \boldsymbol{M_{12}}, \boldsymbol{M_{13}}$, which are given in equations (92), (93), and (94), respectively.

$$\boldsymbol{M_{11}} = \begin{bmatrix} 8 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 5 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 3 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 5 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 4 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 6 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 4 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 4 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 5 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 4 \end{bmatrix} \tag{92}$$

$$\boldsymbol{M_{12}} = \begin{bmatrix} \frac{13}{28} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{3}{5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{3}{5} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{2}{3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{7}{15} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{2}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{5}{6} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2}{5} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2}{3} \end{bmatrix} \tag{93}$$

$$\boldsymbol{M_{13}} = \begin{bmatrix} 0.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.8 \end{bmatrix} \tag{94}$$

5.2.3. Base dataset

A base dataset is formed by the pairs $(\boldsymbol{X_t}, y_t)$, where $\boldsymbol{X_t}$ is derived from equations (89), (90), and (91), and y_t is corresponding exact reliability, for each network t in the set of networks. For instance, considering the sample network $\boldsymbol{X_1}$, the exact reliability y_1 is 0.984263933411563, calculated by using the backtracking algorithm [251]. The base dataset will be used to train and evaluate the CNN, whose architecture is presented in the next section.

5.2.4. Architecture of the proposed CNN

The proposed CNN architecture is based on an image classifier with some appropriate modifications. Several layers of convolutional filters with their respective batch normalization and ReLU layers are proposed based on a generic CNN for image classification. The number of convolutional filters can be investigated for selection of the best number. Furthermore, three important changes are proposed in this study with the aim of predicting all-terminal network reliability. First, the softmax classification output layer needs to be substituted by a regression layer for approximating the target network reliability by minimizing the mean square error. Second, a sigmoid layer with the logistic activation function is placed between the fully connected layer and the output layer. The logistic activation function, as shown in equation (95), allows to ensure the output within the range [0,1], where reliability is defined.

$$g(x) = \frac{1}{1 + e^{-x}} \tag{95}$$

Third, a dropout layer is placed before the fully connected layer. The purpose of the dropout layer is to avoid overfitting by randomly dropping neurons during the training [252]. The probability of a neuron to survive is independent of other neurons and obeys to a given parameter.

The generic proposed architecture is shown in Table 14.

Table 14. Proposed CNN architecture

Topologies Input: ($\boldsymbol{X_t}$)
Several layers of: • Convolution • Batch Normalization • ReLU
Dropout
Fully Connected
Sigmoid Layer
Regression Output: $\widehat{y_t}$

5.2.5. Training and evaluating

Once a base dataset is available and the architecture of the CNN is defined, the remaining phases are training and evaluating. The base dataset is split in training and testing datasets. Several CNNs are obtained based on the proposed architecture by changing the hyperparameters. The CNNs are trained using the training dataset. After training, the CNNs are evaluated in terms of the error using the testing dataset. The process is illustrated in the flowchart as shown in Figure 34.

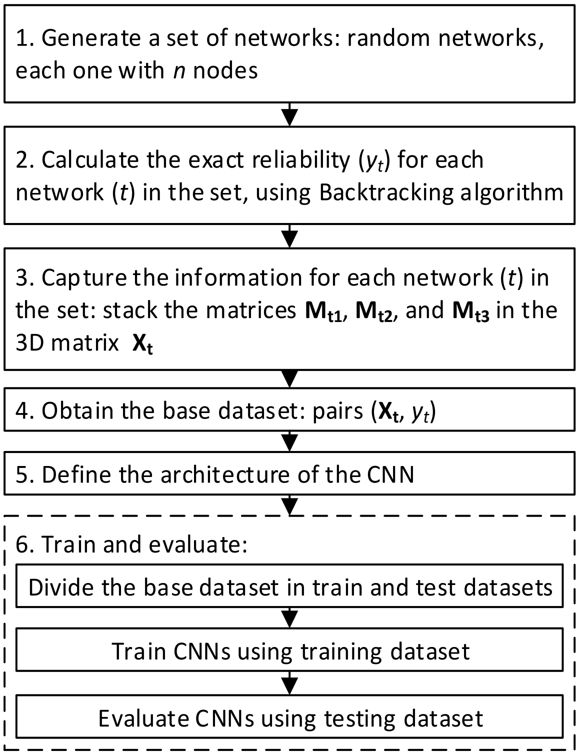

1. Generate a set of networks: random networks, each one with n nodes
2. Calculate the exact reliability (y_t) for each network (t) in the set, using Backtracking algorithm
3. Capture the information for each network (t) in the set: stack the matrices $\mathbf{M_{t1}}$, $\mathbf{M_{t2}}$, and $\mathbf{M_{t3}}$ in the 3D matrix $\mathbf{X_t}$
4. Obtain the base dataset: pairs ($\mathbf{X_t}$, y_t)
5. Define the architecture of the CNN
6. Train and evaluate:
Divide the base dataset in train and test datasets
Train CNNs using training dataset
Evaluate CNNs using testing dataset

Figure 34. Process of reliability estimation using CNNs

5.2.6. Case study

Networks with 10 nodes will be considered, and the link reliability set is $\{0.80, 0.85, 0.90, 0.95, 0.99\}$. We propose to develop a CNN to be trained with a relatively small dataset of different topologies and link reliability values from the given set. In addition to the network topology, topological attributes will be part of the inputs, i.e., the information for each network will be embedded in $\boldsymbol{X_t}$, as discussed earlier. The exact all-terminal network reliability will be the target during the training process, i.e., the y_t values. The trained CNN is expected to estimate the all-terminal network reliability of previously unseen networks. These new unseen networks will have 10 nodes with a link reliability value of either 0.80, 0.85, 0.90, 0.95, or 0.99.

5.2.6.1. Base dataset

In the case study, for ten-node networks$(n = 10)$, a set of 750 random networks is generated. The set size is chosen based on previous studies on all-terminal network reliability estimation using machine learning [42, 45]. This set represents a very small fraction of the number of possible topologies, 1.76×10^{14}. Only networks forming at least one spanning tree are to be considered, i.e., each network in the dataset has a reliability greater than 0. Based on this random set of networks, a base dataset of pairs $(\boldsymbol{X_t}, y_t)$ is formed. The base dataset is divided in training and testing datasets with a proportion of 80% and 20%, a common validation approach in machine learning [42, 45, 253].

The root mean square error (RMSE) was used to evaluate the accuracy of the CNNs in the (unseen) testing dataset.

$$RMSE = \sqrt{\frac{1}{n_{test}} \sum_{i=1}^{n_{test}} (y_i - \widehat{y_\iota})^2} \tag{96}$$

where, n_{test} is the number of elements in the test dataset, y_i is the actual reliability value, and $\widehat{y_\iota}$ is the reliability value predicted by the CNN.

5.2.6.2. Hyperparameters

Different epochs and batch sizes were evaluated during the training. Epochs in the range between 20 and 50 are suggested for image processing with CNN [254, 255], whereas batch sizes in the range between 2 and 32 have been reported to produce the best performance for deep neural networks [256]. Accordingly, in this study, epochs of 30, 40, and 50 along with batch sizes of 8, 16, 32, and 64 were considered, which results in 12 different training configurations.

Four convolutional layers with 8, 16, 32, and 32 filters, respectively were employed. Each convolutional layer was followed by batch normalization and ReLU layers, considering that such a structure generated successful results in a previous image classification study with CNN [254]. In addition to these initial values, different multiplier (*mul*) values of 1, 4, and 8 were applied to the number of filters, i.e., combinations of number of filters (8, 16, 32, 32), (32, 64, 128, 128), and (64, 128, 256, 256) were investigated for selection of the best number of filters. The investigated CNN architecture is shown in Table 15.

Table 15. Investigated CNN architecture – 19 layers

#	Type	Description
1	Topologies Input ($\boldsymbol{X_t}$)	$10 \times 10 \times 3$ input matrices
2	Convolution	8x*mul* filters of size 3x3x3 with stride [1 1] and padding 'same'
3	Batch Normalization	Batch normalization with 8x*mul* channels
4	ReLU	ReLU
5	Average Pooling	2x2 average pooling with stride [2 2] and padding [0 0 0 0]
6	Convolution	16x*mul* filters of size 3x3x(8x*mul*) with stride [1 1] and padding 'same'
7	Batch Normalization	Batch normalization with 16x*mul* channels
8	ReLU	ReLU
9	Average Pooling	2x2 average pooling with stride [2 2] and padding [0 0 0 0]
10	Convolution	32*xmul* filters of size 3x3x(16x*mul*) with stride [1 1] and padding 'same'
11	Batch Normalization	Batch normalization with 32*xmul* channels
12	ReLU	ReLU
13	Convolution	32x*mul* filters of size 3x3x32*xmul* with stride [1 1] and padding 'same'
14	Batch Normalization	Batch normalization with 32x*mul* channels
15	ReLU	ReLU
16	Dropout	Dropout layer
17	Fully Connected	1 fully connected layer
18	Sigmoid Layer	Sigmoid set - range: [0,1]
19	Regression Output	mean-squared-error based

5.2.6.3. Best CNN architecture

The best RMSE value of 0.04406 was obtained for the CNN with a filter multiplier value of 8, and with 30 epochs and an eight-sized batch as training parameters. Indeed, those are the parameters that produced the best RMSE of 0.04406, the measure used to select the best CNN. The proposed CNN outperformed the general ANN proposed by Srivaree-Ratana et al. [42], who reported an RMSE of 0.06260. The better performance might be attributed in part to the multiple hidden layers of the CNN instead of only one hidden layer in the standard ANN architecture [42, 45]. In addition, features such as momentum training [257], dropout, and regularization, are believed to enable higher predictive accuracy compared to typical ANNs [258]. The improvement in accuracy was attained without the need of providing the upper bound reliability as an input to the CNN. Considering upper bound would imply an additional calculation, often with high computational effort [46, 242], for both the training dataset and the new networks presented to the already trained CNN. In addition, in previous studies of network reliability prediction with ANNs, for some points, there were predicted values greater than 1 [42, 45], which, even infrequent, are not realistic. Such errors may lead to wrong calculations based on reliability prediction and subsequently wrong decisions. The CNN architecture presented in this work prevents this by introducing the sigmoid layer.

A paired t-test between the actual reliability and the reliability predicted by the CNN was performed. It resulted in a p-value of 0.2794 and a 95% confidence interval for the mean difference of [-0.0032, 0.0110]. Therefore, there is no significant pairwise difference between actual and the predicted reliability. In addition, Figure 35 shows that the predicted reliability follows the actual reliability reasonably well.

Despite the successful prediction suggested by the RMSE, the p-value, and the good fit (Figure 35), it is worth to notice that the CNN predicted reliability seems to be slightly deviated for extreme low and high reliability values. For instance, in the Figure 35 on the left side, actual network reliability values below 0.93 approximately, are overestimated. This might not be a big concern considering that networks are usually expected to exhibit very high reliability. Moreover, the first 90 samples (plotted on the left side) correspond to link reliability values of 0.80, 0.85, and 0.90, as shown in Figure 36. Such values are unrealistic, as in practical applications, links usually have higher reliabilities [259]. Therefore, they may not be present in a real scenario. Even if those cases of very low true reliability occur, the CNN would be useful. For instance, if at any given moment a network reaches a true reliability of 0.8 which probably be catastrophic, it would be good enough if the CNN can predict a (low) value of say 0.93, which could be used to trigger an alarm or take any appropriate action (manual or automated). It can be seen on Figure 35 that although the prediction performance deteriorates for low "valleys", the predicted reliability curve follows them, i.e., generate valleys as well.

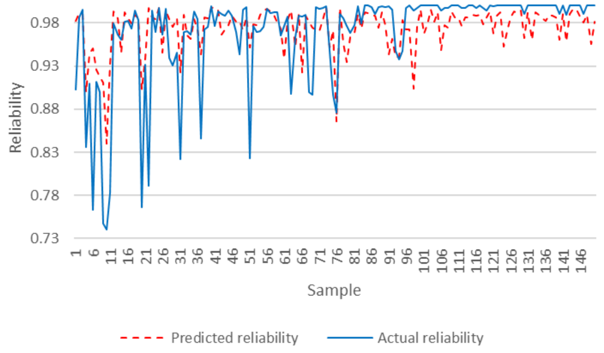


Figure 35. CNN predicted reliability compared to actual reliability for the test dataset

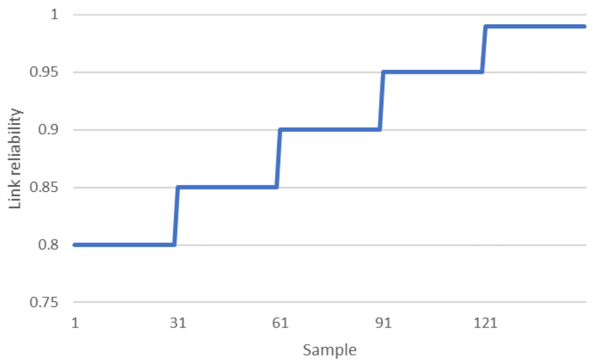


Figure 36. Link reliability of the sample networks in the test dataset

On the other hand, for high values, approximately above 0.98, the CNN seems to underestimate the reliability. For example, if the actual reliability is 0.99, the system might predict 0.98, which will allow to take any appropriate action early (even before the actual reliability drops to 0.98). Similar results were obtained by applying cross-validation, described in the next subsection (5.2.6.4). See for instance, the results for the second fold in Figure 37. The decreased accuracy at extremely low or high reliability values might be addressed, in a future research, by training specialized CNNs for low and/or high reliability values and integrating them in a hierarchical model.

5.2.6.4. Cross-validation

To further investigate the performance of the proposed approach, five-fold cross-validation[253] is considered to compare the results obtained with different test sets from the same base dataset. The base dataset is (randomly) divided into five subsets (150 observations per subset). CNN training uses all but one subset, and the excluded subset is considered a test set for the trained CNN. The best CNN architecture and training parameters are considered for this process, i.e., a filter multiplier of 8, and training parameters batch size of 8 and 30 epochs are employed for each of the five CNNs to be trained and tested using cross-validation. For each fold, the resultant training and testing sets have the same proportion of link reliabilities.

The average RMSE considering cross-validation can be estimated as[42, 253]:

$$RMSE_{cv} = \sqrt{\frac{1}{750}\sum_{g=1}^{5}\sum_{h=1}^{150}\left(y_{(g-1)\times 150+h} - \hat{y}_{(g-1)\times 150+h}\right)^2} \qquad (97)$$

where g is the index of the subset left out, h is the index of the observations in the left-out subset, and the sample. $y_{(g-1)\times 150+h}$ and $\hat{y}_{(g-1)\times 150+h}$ are the actual and predicted reliability values, respectively. RMSE values for each fold as well as average RMSE considering cross-validation are shown in Table 16.

Table 16. RMSE with cross-validation

Fold	RMSE
1	0.04406
2	0.04488
3	0.05953
4	0.05153
5	0.05236
Average	0.05079

The first fold corresponds to the testing dataset obtained from initial division mentioned earlier in section 5.2.6.1. The performance of the CNNs is similar for the other four folds. For instance, the predicted and actual reliability for the second fold is shown in Figure 37, which is similar to Figure 35 (first fold). The average RMSE (0.05079) considering cross-validation is still better than 0.06260, achieved by a regular ANN [42].

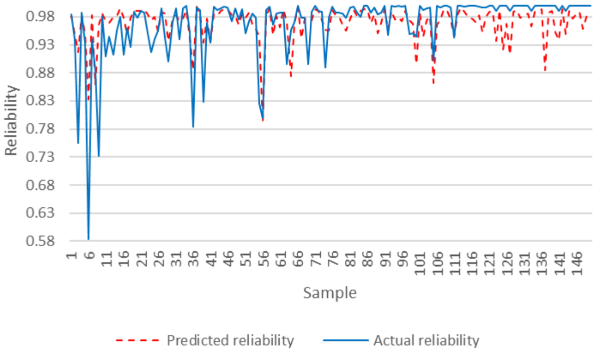


Figure 37. CNN predicted reliability compared to actual reliability for the second fold test dataset

5.2.6.5. Execution time

Although the accuracy is compromised due to the approximation nature of the proposed approach, one of its major contributions is the reduction in calculation time. A PC with a processor Intel(R) Core (TM) i7-6700 CPU @ 3.40GHz, with 16GB in RAM, was used. The time to estimate the reliability with the best CNN was 0.177 seconds for 150 networks, i.e., an average of 1.18 ms/network, whereas the backtracking (exact) algorithm took 299,198 seconds to calculate the reliability for 600 networks, i.e., an average of 499 s/network.

5.2.7. Conclusion

This section has proposed a CNN approach to estimate the reliability of networks based on their topology and topological attributes. The proposed deep learning methodology has generated empirical evidence that supports the hypothesis that the all-terminal reliability of a network can be estimated from its topology and link reliability information embedded as an image-like matrix which can be processed by CNNs. An appropriate format for the inputs and a CNN architecture has been proposed. Several training parameters were evaluated, and the results suggest that 30 epochs and a batch size of 8 are the best combination. In addition, a filter multiplier of 8 seems to be the best value to be considered in the CNN architecture. The best trained CNN proposed here, based on the RMSE, outperforms a previous traditional ANN approach [42], although for the present study the upper bound was not considered as an input. There is a significant computation time reduction achieved by using the proposed CNN.

The proposed approximate method could be used in situations where exact reliability accuracy can be sacrificed to gain calculation speed, such as an online reliability monitoring system. For instance, if one or more links are suddenly broken, the topology of a network will change, and a CNN-based monitoring software would allow to quickly get information on the reliability and take proper action. Another potential use of this method could be during network optimization routines, which require recurrent calculations of networks reliability [42, 131].

5.3. Network reliability estimation with DNN [76]

5.3.1. Introduction

One limitation of previous approaches (including the CNN approach of section 5.2) is that the training and testing processes are specific for a fixed network size, e.g., 10 nodes. Different from previous works, the proposed approach in this section provides a way to predict the reliability of networks with varying graph sizes (i.e., number of vertices or edges). One contribution of this section is integrating several graph embedding methods (GEM) with several DNN architectures for network reliability prediction. This section evaluates six state-of-the-art GEMs with several DNN architectures with varying number of layers. Indeed, the DNN with one hidden layer can be considered a regular ANN. In addition, a dropout layer is placed before the output layer. The purpose of the dropout layer is to avoid overfitting by randomly dropping neurons during the training. The combinations of GEM and DNN architectures are evaluated in function of the error, considering cross-validation. To demonstrate the applicability of the proposed approach, a dataset of 6000 random networks was generated, which is significantly larger than previous datasets of 750 points used with ANNs [42, 45].

5.3.2. Deep neural networks and graph embedding methods

A Deep Neural Network (DNN) can be considered as a stack of multiple hidden layers instead of only one hidden layer in the standard ANN architecture. . A DNN can be used to approximate a function f^*. For instance, $y = f^*(x)$ maps an input $\boldsymbol{x}$ to a real value target y. During the DNN training, the goal is $f(\boldsymbol{x})$ to match $f^*(\boldsymbol{x})$. Graphs that represent networks cannot be directly fed to DNN, because DNN require real vectors as inputs. This study proposes to employ graph embedding methods (GEM) to learn vector representation of graphs, while preserving their properties. Six state-of-the-art GEM will be investigated in this study:

Graph2vec [260], Geometric Scattering for Graph Data Analysis [261], Graph Embedding Enriched by Line Graphs with Edge Features(GL2vec) [262], Family of graph spectral distances (FGSD) [263] Network Laplacian Spectral Descriptor (NetLSD) [264], and Simple and fast algorithm (SF) [265].

5.3.3. Proposed method

This section proposes a DNN approach to estimate the all-terminal reliability of networks with varying graph sizes (i.e., number of vertices or edges) and a given set of possible links reliability values. The proposed model makes the following assumptions for the networks:

1. The nodes are perfectly reliable.
2. The links failure probabilities are independent.
3. The link failure probabilities are equal.

5.3.3.1. Generating the set of networks

A set of Erdős-Rényi graphs is generated, assuming the following: 1) equal proportions of number of nodes from a given set, 2) equal proportions of link reliabilities from a given set, and 3) the probability of edge creation is within a given range $[p_{min} , p_{max}]$.

5.3.3.2. Calculating the exact all-terminal reliability

The exact all-terminal network reliability is calculated using the equation (84), as we did for the CNN approach.

5.3.3.3. Capturing the network information

It is hypothesized in this study that the all-terminal network reliability can be estimated from the graph vector representation learned by the GEMs. This vector representation of the graph will be used as the input for the different DNN architectures.

5.3.3.4. Base dataset

A base dataset is formed by the pairs $(\boldsymbol{X_t}, y_t)$, where $\boldsymbol{X_t}$ is the vector representation generated by GEM, and y_t is corresponding exact reliability, for each network t in the set of networks.

5.3.3.5. Architecture of the proposed DNN

The proposed DNN architecture is based on feed-forward neural networks as they have proven to be effective function approximators [266]. Furthermore, a fully connected multi-layer perceptron (MLP) structure is employed. Rectified linear unit activation function (ReLU) is used as it is the default recommendation in modern neural networks. A hyperbolic tangent activation function is used at the output layer to ensure the reliability predicted falls within the range [0,1].

In addition, a dropout layer is placed before the output layer to avoid overfitting by randomly dropping neurons during the training. The generic proposed architecture is shown in Table 17.

Table 17. Proposed DNN architecture

Topologies Input: $(\boldsymbol{X_t})$
Fully connected layers with ReLU activation
Dropout
Output with tanh activation: $\widehat{y_t}$

5.3.3.6. Training and evaluating

Once a base dataset is available and the architecture of the DNN is defined, the remaining phases are training and evaluating. Ten-fold cross-validation is considered to compare the results obtained with different test sets from the same base dataset. The base dataset is (randomly) divided into ten subsets of the same length. DNN training uses all but one subset, and the excluded subset is considered a test set for the trained DNN.

After training, the DNNs are evaluated in terms of the error using the testing datasets from cross-validation. The flowchart in Figure 38 illustrates the process.

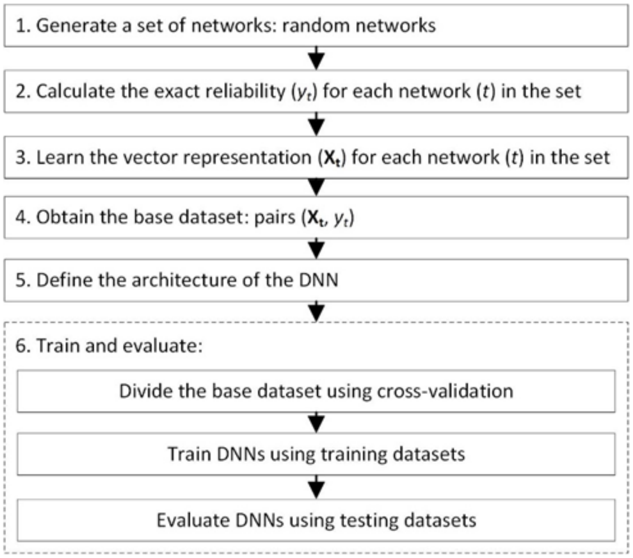

1. Generate a set of networks: random networks
2. Calculate the exact reliability (y_t) for each network (t) in the set
3. Learn the vector representation ($\mathbf{X_t}$) for each network (t) in the set
4. Obtain the base dataset: pairs ($\mathbf{X_t}$, y_t)
5. Define the architecture of the DNN
6. Train and evaluate:
Divide the base dataset using cross-validation
Train DNNs using training datasets
Evaluate DNNs using testing datasets

Figure 38. Process of reliability estimation using DNNs

5.3.4. Case study

Networks with 8, 9, and 10 nodes will be randomly generated. The link reliability set is {0.80, 0.85, 0.90, 0.95, 0.99}, and the probability of edge creation is in the range $[0.3\ ,0.6]$. The information for each network will be embedded in a vector representation $\boldsymbol{X_t}$, using the different GEM methods. The exact all-terminal network reliability will be the target during the training process, i.e., the y_t values. The trained DNN is expected to estimate the all-terminal network reliability of previously unseen networks. These new unseen networks will have either 8, 9, or 10 nodes with a link reliability value of either 0.80, 0.85, 0.90, 0.95, or 0.99, and edge density between 0.3, and 0.6. Python was used for both network dataset generation and DNN experiments.

5.3.4.1. Base dataset

A dataset of 6000 networks is generated considering only connected graphs, i.e., each network in the dataset has a reliability greater than 0. Based on this random set of networks, a base dataset of pairs $(\boldsymbol{X_t}, y_t)$ is formed. The base dataset is divided in training and testing datasets by applying ten-fold cross-validation.

5.3.4.2. Experiments

A number of hidden layers of one, two, and three is considered to investigate different depths. Dropout probability of 0 (no dropout), 0.25, and 0.5 are explored as well. This provides a total of nine experiments. Each experiment is performed for vector representations of the input graphs for each of the graph embedding methods. The number of nodes of the first hidden layer is set to 128 to match the default dimensionality of the vectors generated by the GEM. The subsequent layers are reduced by a factor of 50% [267]. Experiment 1 corresponds to a traditional ANN architecture, which is useful for comparison.

Table 18. Experiments

Experiment	Hidden layers	Dropout
1	One: 128 neurons	0
2	Two: 128, 64 neurons	0
3	Three: 128, 64, 32 neurons	0
4	One: 128 neurons	0.25
5	Two: 128, 64 neurons	0.25
6	Three: 128, 64, 32 neurons	0.25
7	One: 128 neurons	0.5
8	Two: 128, 64 neurons	0.5
9	Three: 128, 64, 32 neurons	0.5

5.3.4.3. Error

Table 19. RMSE results

Exp.	Graph 2vec	Geo. Scatt.	GL 2vec	FGSD	Net LSD	SF
1	0.14330	*0.01678*	0.12818	0.08723	0.10552	0.09056
2	0.14600	*0.01287*	0.11325	0.08547	0.09594	0.08697
3	0.14511	***0.01069***	0.11582	0.08579	0.09488	0.08724
4	0.13890	*0.02639*	0.12116	0.08695	0.10550	0.09273
5	0.14656	*0.01598*	0.11675	0.08638	0.09850	0.08789
6	0.14702	*0.01781*	0.11542	0.08627	0.09474	0.08739
7	0.13895	*0.02936*	0.11764	0.08759	0.11044	0.09476
8	0.14730	*0.02174*	0.11585	0.08700	0.10166	0.08856
9	0.14853	*0.02242*	0.11768	0.08700	0.09663	0.08754

The root mean square error (RMSE), for each fold is given by equation (96), by letting n_{test} be the number of elements in the test dataset, y_i the actual reliability value, and $\hat{y_i}$ the reliability value predicted by the DNN. Similarly, the average RMSE considering cross-validation can be estimated as [42]:

$$RMSE_{cv=}\sqrt{\frac{1}{6000}\sum_{g=1}^{10}\sum_{h=1}^{600}\left(y_{(g-1)\times 600+h}-\hat{y}_{(g-1)\times 600+h}\right)^2} \tag{98}$$

where g is the index of the subset left out, h is the index of the observations in the left-out subset, and the sample. $y_{(g-1)\times 600+h}$ and $\hat{y}_{(g-1)\times 600+h}$ are the actual and predicted reliability values, respectively.

The results of the RMSE for the nine experiments are shown in Table 19. The minimum RMSE for each GEM (column) is underlined, whereas the minimum RMSE for each experiment (row) is in Italic. The best RMSE value of 0.01069 was obtained for the DNN with 3 hidden layers and no dropout using geometric scattering embedding. Interestingly, for all the nine experiments, the geometric scattering embedding produces the best results. This could be explained because this GEM utilizes topological attributes like the clustering coefficient, which is considered a measure of the disconnectedness of a graph [249], and disconnectedness impacts the reliability as shown in equation (81). The enhanced performance produced by geometric scattering embedding is believed to hold for all different networks' topologies, which result from the varying studied parameters (e.g., number of nodes, and link density). This assumption seems plausible since the number of nodes, link reliabilities, and link densities were taken in uniform proportions to generate the (ten) testing datasets. It is also worth to notice that the traditional ANN approach (experiment 1) was outperformed by DNNs, as none of the experiment 1 results are the minimum for any of the GEMs. It is remarkable that DNNs without dropout achieved the minimum RMSE with four (out of six) GEMs: geometric scattering, GL2vec, FGSD, and SF. Moreover, 3 of such DNNs correspond to the architecture with 2 hidden layers (experiment 2). A dropout of 0.25 produced the minimum only for two GEMs: graph2vec and NetLSD. On the other hand, the dropout of 0.5 did not produce any minimum, which suggest that this value is too large for this application.

The best DNN outperformed previous ANN-based approaches, e.g., RMSE of 0.06260 [42] and RMSE of 0.01878 [45]. The enhanced performance of the proposed approach might be attributed in part to the multiple hidden layers of the DNN instead of only one hidden layer in the standard ANN architecture [42, 45]. In addition, features such as momentum training and regularization, are believed to enable higher predictive accuracy compared to typical ANNs. Moreover, the geometric scattering method seems to capture the topological attributes relevant to the reliability calculation. The improvement in accuracy was attained without the need of providing the upper bound reliability as an additional input, as in previous studies. In addition, in previous studies of network reliability prediction with ANNs, for some points, there were predicted values greater than 1 [42, 45], which, even infrequent, are not realistic. Such errors may lead to wrong calculations based on reliability prediction and subsequently wrong decisions. The DNN architecture presented in this work prevents this by utilizing the hyperbolic tangent activation function. In addition to the RMSE metric, a paired t-test between the actual reliability and the reliability predicted by the DNN was performed. The null hypothesis is that the difference between the actual reliability and the predicted has a mean equal to zero. The test resulted in a p-value of 0.0820 and a 95% confidence interval for the mean difference of [-0.00051, 0.00003]. The p-value greater than 0.05 suggests that there is not enough evidence to reject the null hypothesis. Therefore, there is not significant pairwise difference between actual and the predicted reliability. Furthermore, Figure 39 shows a that the predicted reliability curve follows the actual reliability.

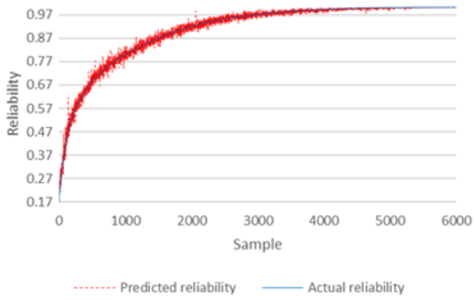


Figure 39. DNN predicted reliability vs. actual reliability

5.3.4.4. Execution time

Although the accuracy is compromised due to the approximation nature of the proposed approach, one of its major contributions is the reduction in calculation time. A PC with a processor Intel(R) Core (TM) i7-6700 CPU @ 3.40GHz, with 16GB in RAM, was used. The average time to estimate the reliability with the best DNN (and geo. scattering) was approximately 3 ms/network, whereas the exact algorithm took an average of 66 s/network.

5.3.5. Conclusion

This section has proposed an approach based on the integration of DNN and GEM to estimate the reliability of varying size networks accurately and quickly.

CHAPTER 6. ALL-TERMINAL NETWORK RELIABILITY ASSESSMENT AND UPDATING OF PARAMETERS CONSIDERING DEGRADATION OF LINKS AND NODES[4]

6.1. Abstract

In most of previous research where reliability of complex CPS is modeled through networks, nodes are considered to have a binary state, i.e., functioning or failed. In this chapter we seek modeling a CPS by considering a network, whose nodes and edges degrade. As an initial approximation to the problem, in CHAPTER 5 only link failures were be considered in the reliability estimation of networks using deep learning approaches. On the other hand, in this chapter we relax the assumption of perfect nodes. Section 6.2 is based on our paper presented at *IISE 2021 Annual Conference* [77] and presents a DNN-MC approach to estimate the reliability of a network considering that both links and nodes can fail. Moreover, in section 6.3, a framework to estimate the reliability of a network considering not only failure but degradation of both links and nodes will be proposed. The proposed framework allows estimating the reliability of a network as a function of time, not only as point estimates but also as credible intervals. The

[4] The present chapter is based on the following papers:

1. A. Davila-Frias, N. Yodo, T. Le, and O. P. Yadav, " A Deep Neural Network and Bayesian Method based Framework for All-Terminal Network Reliability Estimation Considering Degradation" 2021. In Reliability Engineering and System Safety. Submitted.
   Contribution of Alex Davila Frias: developing the mathematical models, analysis of the case study, discussion of the results, and drafting the paper.
   Contribution of Nita Yodo, Trung Le, and Om Yadav: verification of the results and proofreading the draft paper.

2. A. Davila-Frias, O. P. Yadav, S. Salem, and B. Nepal, "All-Terminal Network Reliability Estimation with Monte Carlo and Deep Neural Networks," 2021. In IISE 2021 Annual Conference. Presented.
   Contribution of Alex Davila Frias: developing the mathematical models, analysis of the case study, discussion of the results, and drafting the paper.
   Contribution of Om Yadav, Saeed Salem, and Bimal Nepal: verification of the results and proofreading the draft paper.

proposed framework integrates BM, MC, and DNN, and is capable of incorporate new data, as they become available, for updating of both model parameters and reliability predictions. This section is based on our paper submitted to the journal *Reliability Engineering and System Safety* [78].

6.2. All-terminal network reliability estimation with Monte Carlo and deep neural networks [77]

6.2.1. Introduction

In most of the previous works on network reliability, only links have been considered to fail [40, 42, 45, 55, 75] and nodes have been considered perfect. However, in reality both kind of components, i.e., nodes and links, may fail. Even considering link failures only, the problem is complex and NP hard. Therefore, to overcome the complexity of network reliability estimation, traditional artificial neural networks (ANNs) have received attention recently [42, 45, 75]. ANNs have evolved to deep learning (DL) approaches such as deep neural networks (DNNs), convolutional neural networks (CNNs), and recurrent neural network (RNNs). These advanced DL methods have been utilized in the reliability estimation problem. Although DL has been applied for components reliability estimation, little evidence is available of its use for network reliability estimation. Moreover, in previous studies based on deep learning techniques, the exact reliability was used to train specialized ANNs. This algorithm, although exact, is time consuming and might not be practical for networks with more than ten nodes [42, 75]. Due to the complexity of the problem, and as an alternative to get fast estimations of the reliability of a network, an integration of Monte Carlo and DNNs is proposed. The aim of this study is to develop a framework to address the problem of network reliability estimation considering imperfect links and nodes. A method based on a Monte Carlo algorithm is proposed that can provide estimation of network reliability for given nodes and links reliability values. To speed-up the calculation, a DNN is integrated into the framework. This work evaluates different architectures proposed for a DNN. The proposed architectures consider the sigmoid activation function, which will prevent predicting values out of the range as some obtained by ANNs in previous works [42, 45]. In addition, a dropout layer is placed before the output layer. The purpose of the dropout layer is to avoid overfitting by randomly dropping neurons during the training. The architectures are evaluated in function of the error, considering cross-validation. To demonstrate the applicability of the proposed approach, two complete networks with five and ten nodes, respectively, were generated. In addition, four real-world networks, with number of nodes ranging from 20 to 158, were analyzed using the proposed framework.

6.2.2. Proposed method

This section proposes a framework to estimate the all-terminal reliability of a network considering imperfect links and nodes. To this end, a network is modeled by a graph $G(N, L, p_L, p_N)$, where N is the set of nodes, L is the set of links, p_L is the links reliability, and p_N is the nodes reliability. The proposed approach makes the following assumptions for the network:

1. The links failure probabilities are independent.
2. The link failure probabilities are equal.
3. Similarly, the nodes failure probabilities are equal and independent.
4. The network has bi-directional links.

The proposed methodology involves four broad steps: 1) Estimate the all-terminal reliability with a method based on a Monte Carlo algorithm for a set of possible links and nodes reliability values, to obtain a dataset 2) Define possible architectures of DNNs. 3) Train and evaluate the DNNs considering cross-validation. 4) Calculate the network reliability.

6.2.2.1. Estimating the all-terminal reliability with a Monte Carlo algorithm

At any time, only some links and/or nodes of G might be operational. Since the all-terminal reliability is the probability that every node can communicate with every other node in the network, the reliability of a network is given as:

$$R_{net} = \Pr\{(Operational\ links\ connect\ all\ nodes) AND\ (all\ nodes\ operational)\} \tag{99}$$

Even considering only link failures, and due to the exponential growth of number of states with the size of networks, the all-terminal reliability calculation is an NP-hard problem [241]. In previous studies based on deep learning techniques, the exact reliability was used to train specialized ANNs. This algorithm, although exact, is time consuming and might not be practical for networks with more than ten nodes [75]. For instance, the backtracking exact algorithm took on average about 500 seconds per network [75] on ten-node networks.

As an alternative, a Monte Carlo method could be used to estimate the reliability of a network (R_{MC}). The algorithm should simulate M states (replication) for the network. For each replication, the algorithm should simulate the nodes, considering the reliability of nodes p_N. If not all the nodes are present, then that state is not operational because there is not all-terminal communication. If all the nodes are present, the algorithm should simulate the links considering the reliability of links p_L. If the operational links provide all-terminal connectivity, then that state is operational for the network, and the accumulator variable r (with initial value of zero) is increased by one. After M replications, the estimated reliability would be given by the ratio of the number simulated operational states over the number of simulated states, i.e.:

$$\hat{R}_{net} = R_{MC} = r/M \tag{100}$$

From equation (99) and assumptions 1 and 3, a more efficient way to estimate the network reliability can be derived. Given that the links and nodes reliability values are independent, equation (99) can be expressed as:

$$R_{net} = \Pr\{(Operational\ links\ connect\ all\ nodes)\} \times \Pr\{(all\ nodes\ operational)\} \qquad (101)$$

Let , $R_{links} = Pr\{(Operational\ links\ connect\ all\ nodes)\}$, and $R_{nodes} = Pr\{(all\ nodes\ operational)\}$.

Therefore, the reliability of the network can be expressed as the product of the reliability of links (R_{links}) and reliability of nodes (R_{nodes}) as:

$$R_{net} = R_{links} \times R_{nodes} \qquad (102)$$

The reliability of nodes can be directly calculated by using the probability mass function (PMF) of binomial distribution $Bin(|N|, p_N)$ for $|N|$ successes. In other words, the reliability of nodes is given by equation (103). Consequently, the simulation of nodes is not required.

$$R_{nodes} = p_N^{|N|} \qquad (103)$$

On the other hand, regarding the reliability of links, at any time, only some links of G might be operational. A state of G is a sub-graph $G' = (N, L')$, where L' is the set of operational links, $L' \subseteq$ L. The reliability of links of state $L' \subseteq$ L is:

$$R_{links} = \sum_{\Omega} \left[\prod_{j \in L'} p_L \right] \left[\prod_{j \in (L \setminus L')} (1 - p_L) \right] \qquad (104)$$

where, Ω is the set of all operational states. As mentioned before, due to the exponential growth of number of states with the size of network, the all-terminal reliability calculation is an NP-hard problem. Hence, only the reliability of links needs to be simulated with a Monte Carlo algorithm, which is proposed below:

Algorithm 6.1

```
Let $M$ be the total number of independent replications for Monte Carlo simulation
Let $|N|$ be the number of elements (nodes) in $N$
$r \leftarrow 0$
$k \leftarrow 0$
while $k < M$ do
$L_k \leftarrow L$
        //Simulate the links …
        for each $l_j \in L_k$ do
generate a random number $rep_{l_j}$ from $Bernoulli(p_L)$
if $rep_{l_j} = 0$ then remove $l_j$ from $L_k$
        //Check connectivity …
if all-node connectivity in $G(N, L_k)$ then $r \leftarrow r + 1$
        $k \leftarrow k + 1$
$R_{MC_{links}} = r/M$
```

By using the results of Algorithm 6.1 and equation (103), the network reliability can be estimated as:

$$\hat{R}_{net} = R_{MC_{links}} \times R_{nodes} \tag{105}$$

where $R_{MC_{links}}$ is evaluated by algorithm 6.1 and R_{nodes} is calculated with equation (103).

A base dataset is formed by the pairs (X_t, y_t), where X_t is the links reliability value (p_L), and y_t is corresponding estimated reliability of links $R_{MC_{links}}$, for each element in a set of link reliability values.

6.2.2.2. Architecture of DNN

The proposed DNN architecture is based on feed-forward neural networks as they have proven to be effective function approximators [266]. Furthermore, a fully connected multi-layer perceptron (MLP) structure is employed. A sigmoid activation function is used at the output layer to ensure the reliability predicted falls within the range [0,1].

In addition, a dropout layer is placed before the output layer to avoid overfitting by randomly dropping neurons during the training. To summarize, the architecture is a stack of the following layers: input: (X_t), fully connected hidden layers, dropout, output with sigmoid activation: $\widehat{y_t}$.

6.2.2.3. Training and evaluating

Once a base dataset is available and the architecture of the DNN is defined, the remaining phases are training and evaluating. Five-fold cross-validation is considered to compare the results obtained with different test sets from the same base dataset. The base dataset is (randomly) divided into five subsets of the same size. DNN training uses all but one subset, and the excluded subset is considered a test set for the trained DNN. After training, the DNNs are evaluated in terms of the error using the testing datasets from cross-validation.

6.2.2.4. Calculating the network reliability

The cross-validation error is used to select the best DNN. The best trained DNN can be used to estimate the reliability of links ($\hat{R}_{links}$) for a given links reliability value (p_L). Further, the reliability of nodes (R_{nodes}) for a given nodes reliability value (p_N) is given by equation (103). Therefore, the estimated reliability of the network is given as:

$$\hat{R}_{net} = \underbrace{DNN(p_L)}_{\hat{R}_{links}} \times R_{nodes}(p_N) = DNN(p_L) \times p_N^{|N|} \tag{106}$$

6.2.3. Case study

Six networks are evaluated to demonstrate the applicability of the proposed framework: (a) a synthetic five-nodes complete (10-links) network, (b) a synthetic ten-nodes complete (45-links) network, (c) Arpanet (20 nodes, 32 links) [268], (d) Intellifiber (97 nodes, 73 links, USA), (e) Ion (125 nodes, 150 links, New York, USA), (f) US Carrier (158 nodes, 189 links, Georgia, Alabama, Florida, USA) [269]. Figure 40 shows the graphical representation of the six networks considered for the case study.

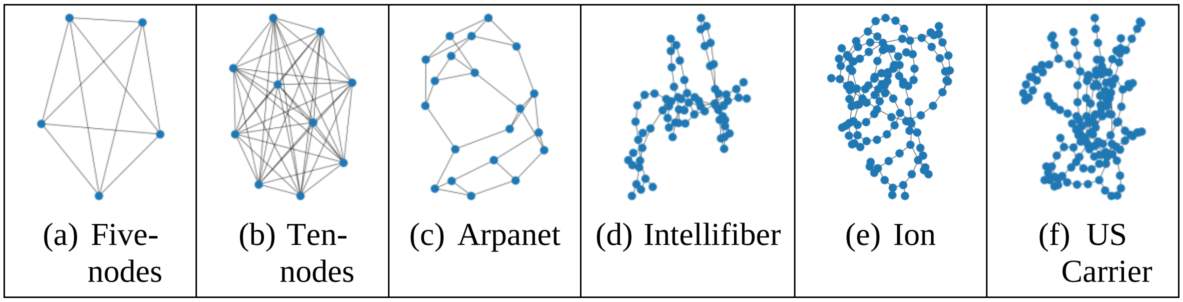

(a) Five-nodes (b) Ten-nodes (c) Arpanet (d) Intellifiber (e) Ion (f) US Carrier

Figure 40. Networks representation

The estimated reliability of links ($R_{MC_{links}}$), will be the target during the training process, i.e., the y_t values. The best trained DNN for each network is expected to estimate the network reliability of links ($\hat{R}_{links}$) for any given value of p_L. The all-terminal network reliability will be calculated by using equation (106) for any given p_L and p_N.

6.2.3.1. Base dataset

For each network, a dataset of 100 link reliability values is considered {0.01, 0.02, …, 0.99, 1.00}. Based on this set of link reliability values, a base dataset of pairs (X_t, y_t) is formed. The base dataset is divided in training and testing datasets by applying five-fold cross-validation.

6.2.3.2. Experiments

The DNN architecture has two hidden layers, and we investigate different number of neurons ({5, 10, 20, 30, 40, 50}) for each hidden layer [267]. The dropout probability values from the set {0, 0.05, 0.10, 0.15, 0.20, 0.25}, where 0 indicates no dropout is employed. This provides a total of 216 experiments, performed for each of the six networks of the case study. The average root mean square error (RMSE) considering cross-validation is used to select the best DNN architecture. The best architecture for each network is shown in Table 20 (column 2). The architecture is expressed as the number of neurons in the first hidden layer, number of neurons in the second hidden layer, and dropout probability. For instance, (50, 50, 0) means 50 neurons, 50 neurons, and 0 dropout probability. The final application DNN is trained using all the 100 members of the data set and its validation error is inferred using the average cross-validation error (Table 20, column 3). The average cross validation-error is given by equation (107) [42, 75].

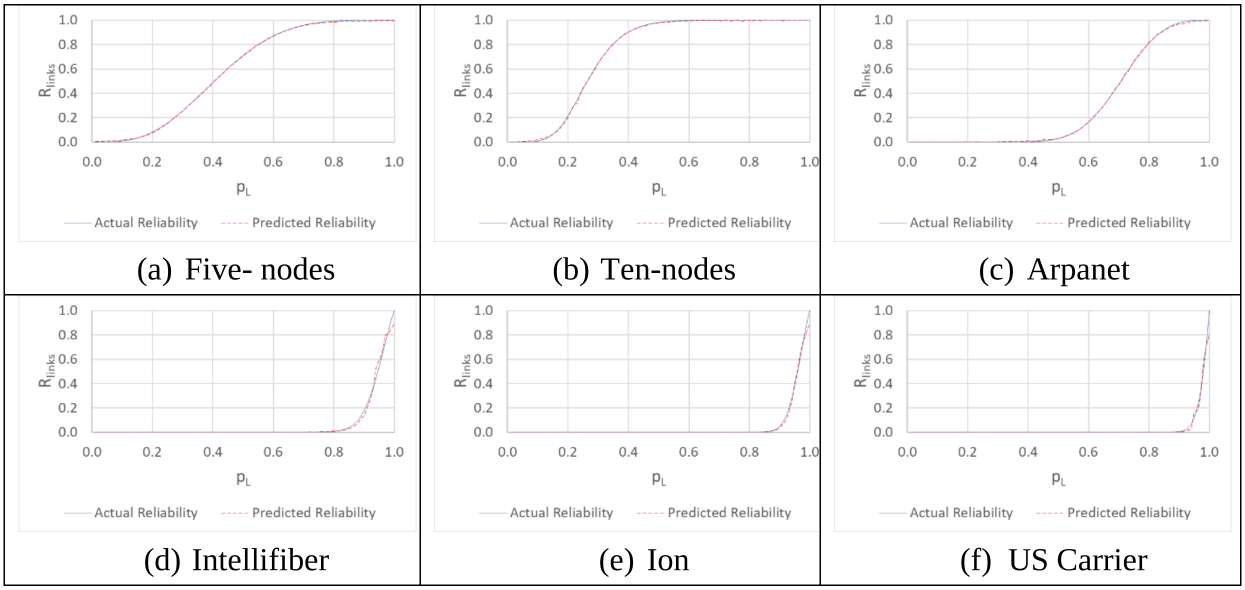
$$RMSE_{cv=}\sqrt{\frac{1}{100}\sum_{g=1}^{5}\sum_{h=1}^{20}\left(y_{(g-1)\times 20+h}-\hat{y}_{(g-1)\times 20+h}\right)^{2}} \tag{107}$$

Table 20. Summary of results

	Architecture	Error	Paired t test		Computation time	
Network	Best DNN	RMSE	p-value	95% C.I.	Monte Carlo [s]	MC-DNN [ns]
a	50, 10, 0	0.00351	0.0820	[-0.0001, 0.0013]	60.73	39.96
b	50, 5, 0	0.00591	0.7136	[-0.0010, 0.0014]	161.23	41.56
c	50, 10, 0	0.00477	0.9453	[-0.0010, 0.0009]	135.05	40.22
d	5, 5, 0.1	0.01907	0.3981	[-0.0022, 0.0054]	790.98	46.45
e	5, 30, 0.15	0.01460	0.1029	[-0.0005, 0.0053]	1223.59	46.43
f	20, 10, 0.25	0.02213	0.2728	[-0.0019, 0.0068]	740.41	46.49

There is a significant computation time difference between the pure Monte Carlo algorithm and the integrated framework based on Monte Carlo-DNN (Table 20, columns 6, 7). The Monte Carlo average time is based on 100 link reliability values. The DNN average computation time is estimated by dividing the time to compute the reliability of a $10^4 \times 10^4$

array by 10^8 (Used to plot Figure 42). In addition, the pure Monte Carlo method provides the reliability for a specific links reliability value (p_L). On the other hand, the proposed framework estimates the network reliability for any given links reliability (p_L) and nodes reliability (p_N) values. The estimated network reliability for the combination of 10,000 values for both p_L and p_N uniformly distributed between 0 and 1 is plotted in Figure 42. A PC with a processor Intel(R) Core (TM) i7-6700 CPU @ 3.40GHz, and 16GB in RAM was used.



(a) Five- nodes

(b) Ten-nodes

(c) Arpanet

(d) Intellifiber

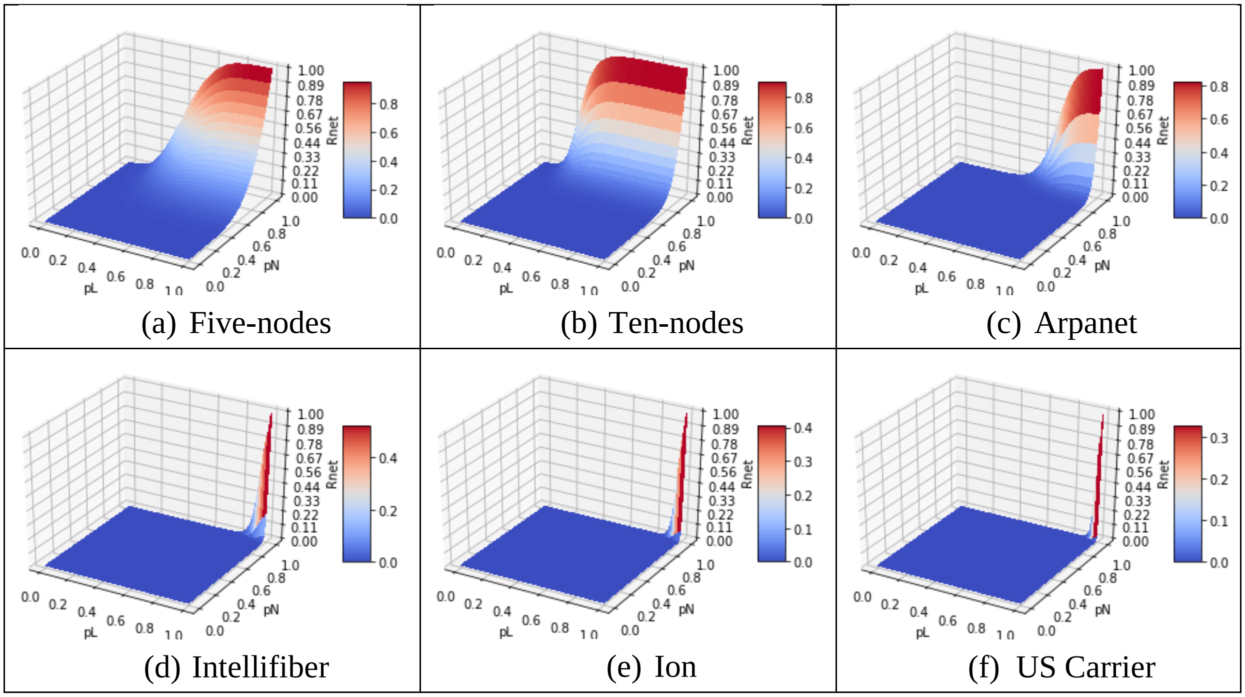
(e) Ion

(f) US Carrier

Figure 41. Estimated reliability of links ($\hat{R}_{links}$), as a function of links reliability (p_L)

The error (0.02213 in the worst case) is comparable with previous results achieved by ANN-based approaches, e.g., RMSE of 0.06260 [42] and RMSE of 0.04406 [75]. Also, a paired t-test between the actual reliability and the reliability predicted by the DNN was performed. p-values and 95% confidence intervals (Table 20, columns 4, 5) for the mean difference show no significant pairwise difference between actual and the predicted values. Therefore, the DNN provides good fit, as shown in Figure 41. For the largest networks (d, e, and f), the prediction for links reliability values greater than 0.99 approximately underestimate the actual reliability. To improve this performance, a hierarchical approach that integrates specialized DNNs trained for link reliabilities greater than 0.99, is used. The best specialized DNN architectures were (50, 50, 0) for both networks d, and e, and (40, 20, 0) for network f. The hierarchical approach allows a smooth fit even at high reliability values, as shown in Figure 42.

Figure 42 shows that, in general, the nodes reliability is more dominant than the links reliability, which can be explained because a failure in a node immediately interrupts the all-terminal communication, whereas a link failure may be alleviated by communication trough other surviving links. In addition, larger networks (d, e, f) are less sensitive to low values of links reliability p_L, which can be explained by a larger number of links providing alternative communication paths when other links fail.



(a) Five-nodes

(b) Ten-nodes

(c) Arpanet

(d) Intellifiber

(e) Ion

(f) US Carrier

Figure 42. Estimated reliability of networks ($\hat{R}_{net}$), as a function of links reliability (p_L) and nodes reliability (p_N)

6.2.4. Conclusions and discussions

A framework was proposed for fast reliability estimation of large networks considering imperfect links and nodes. Exact reliability algorithms may not be practical for networks with more than ten nodes. Therefore, to estimate the network reliability, a method based on Monte

Carlo – DNN is proposed, which leads to a significant computation time reduction. The framework performs reliability estimation in less than 100 ns for any given links and nodes reliability values, using a regular desktop PC. The results show that the nodes reliability is more dominating than the links reliability. The proposed framework can be used for real-time monitoring of network reliability, provided there is information of the links reliability and nodes reliability, which could be captured from degradation data collected by modern sensor technology.

6.3. All-terminal network reliability estimation considering degradation with Bayesian methods, Monte Carlo, and deep neural networks [78]

6.3.1. Abstract

In most of previous research on network reliability, links are considered to have a binary state, i.e., functioning or failed, whereas nodes are considered perfect. In a more realistic scenario, both links and nodes might fail. Moreover, links and nodes may exhibit degradation behavior before failing. This study develops a framework to estimate the all-terminal reliability of a network, whose nodes and links not only have the possibility to fail but also exhibit degradation. Different from previous works on network reliability that have considered constant reliability for links, this study considers the reliability of links, nodes, and network as functions of time. To this end Bayesian methods (BM) are proposed to estimate reliability of links and nodes as functions of time considering degradation data. Due to the complexity of the all-terminal reliability problem, and to get fast estimations of the reliability of a network, an integration of Monte Carlo (MC) and Deep Neural Networks (DNNs) is proposed. The proposed Monte Carlo-based algorithm can provide estimation of the network reliability for given nodes and links reliability values. To speed-up the calculation, a DNN model is integrated into the

framework, thus enabling accurate and fast estimation of network reliability for given link and node reliability values. The DNN accuracy, based on the RMSE (0.01460), outperforms previous traditional artificial neural network (ANN) approaches. Moreover, the DNN model takes 0.3 ms to compute the reliability for any given links and reliability values. The framework can provide not only reliability point estimates, but also credible intervals. Finally, we take advantage of Bayesian methods to integrate new data to the framework as they become available. The new data are used by the framework to refine and further update the degradation model parameters and the prediction of reliability of links, nodes, and the network. The proposed methodology has been demonstrated with the real-world network topology Ion (125 nodes, 150 links) considering real degradation data.

6.3.2. Introduction

In previous works on network reliability, binary states have been commonly assumed for links, and nodes have been considered perfect [40, 42, 45, 55, 131]. Even considering link failures only, the problem is complex and NP hard [40, 42]. Traditional network reliability methods include exact NP-hard methods [32, 39, 40, 129] or approximated methods. Among approximated methods, there are several methods such as graph reduction [39, 40], cut-set and tie-set approximations [40], Monte Carlo (MC) [30, 47-49, 270] and bounds [40, 50, 51]. On the other hand, among modern approximated approaches based on deep learning, artificial neural networks (ANNs) have emerged as a promissory tool to estimate network reliability. Indeed, ANNs have been claimed to be one of the most efficient methodologies developed so far for reliability estimation of networks [130].

ANNs have been usually trained with the network topology and link reliability as inputs and with the target network reliability as desired output [42, 45, 131]. For example, Srivaree-

Ratana et al. [42] utilized an ANN to predict the all-terminal network reliability; with the network architecture, the link reliability, and the network reliability upper bound (an approximation of network reliability which is not lower than the exact value [40, 46, 242]) as inputs, and the exact network reliability as the target. More recently, Altiparmak et al. [45] proposed an ANN model to predict the all-terminal network reliability, which takes the node degree and other connectivity metrics and the upper bound network reliability as inputs to predict the network reliability. Traditional ANNs have evolved to deep learning (DL) approaches [271, 272] such as deep neural networks (DNNs), convolutional neural networks (CNNs), and recurrent neural network (RNNs). These advanced DL methods have been utilized in the reliability estimation problem. For instance, RNNs have been successfully applied to predict health and remaining useful life of bearings [273], li-ion batteries [274]. Also, CNNs have been applied to evaluate online services reliability [275], software reliability [276] a robot's pose and reliability [277], rotating machinery reliability [278], and recently, network reliability [75]. Similarly, a DNN has been used for health prognostic of li-ion batteries [258, 279] and RUL of bearing [280]. Although DNNs have been applied for reliability estimation, little evidence is available of its use for network reliability estimation.

Both traditional approximated methods like MC, and modern techniques, such as those based on ANN, have mostly considered link failures possibility only. However, in reality, both kind of components, i.e., nodes and links, may fail. Moreover, both links and nodes may not only fail but exhibit degradation with time. Such degradation can provide useful information to estimate the reliability of both links and nodes of a network as functions of time. Whereas most of the current work on network reliability considers that the reliability of links or nodes is constant or even perfect, the aim of this study is to consider the reliability of links and nodes as a function of time in the prediction of all-terminal network reliability. One major contribution of this work is to integrate the concepts of component reliability based on degradation data, and network reliability, by modeling components (as nodes and links) that degrade. As a result, this work provides a framework to estimate the network (all-terminal) reliability as an indicator of the overall health condition of the network. This work proposes a framework to use degradation data from both links and nodes of a network to estimate its all-terminal reliability as a function of time, to account for its dynamic behavior. Due to the complexity of the problem, the proposed framework integrates BM, MC simulation, and DNNs. BM with low-information prior distributions are proposed to estimate the degradation parameters and to further evaluate the reliability (as functions of time) of links and nodes from degradation data. As a second step, a method based on a MC algorithm is proposed that can provide estimation of network reliability function for given links and nodes reliability functions. Since, the MC method provides good estimates but is time consuming, a DNN model is designed and integrated into the framework to speed-up the calculation. The DNN model is trained for a range of links and nodes reliability values to learn the all-terminal network reliability calculated with the MC method. In addition, the framework allows to incorporate new data, as they become available, to update the reliability predictions of links, nodes, and network. Moreover, the proposed framework provides not only point estimates, but also Bayesian credible intervals for the reliability functions of links, nodes, as well as for all-terminal network reliability function. To demonstrate the applicability of the proposed approach, the real-world network topology Ion (125 nodes, 150 links, New York, USA) [269, 281] was analyzed using the proposed framework. Real degradation data was considered for nodes [282]. Similarly, data was simulated based on real degradation data for links [59].

6.3.3. Proposed methodology

This section presents the proposed framework for all-terminal reliability function estimation of a network considering degradation of its links and nodes. Different from previous works, we relax the perfect nodes assumption. We propose to model a network by a graph $G(N, L, p_L, p_N)$, where N is the set of nodes, L is the set of links, p_L is the links reliability, and p_N is the nodes reliability. For a given network, the reliability values p_L and p_N in reality are not constant, as both links and nodes may not only fail but degrade with time. Therefore, such values can be considered as functions of time that can be calculated from degradation data of the links and nodes, respectively. This section considers nodes and links as sample units from two populations because nodes and links represent different types of components and may exhibit different degradation profiles. A representation of a network with degradation in links and nodes is shown in Figure 43. Degradation data from both links and nodes will be used to estimate the reliability values p_L, and p_N as functions of time, respectively. Degradation patterns for links are represented by solid curves, whereas degradation for nodes is symbolized by the dashed curves.

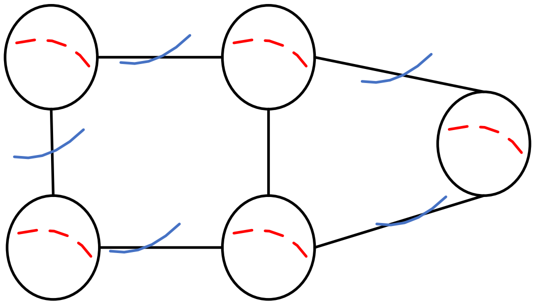


Figure 43. Representation of a network with degradation in links and nodes

The proposed approach makes the following assumptions for the network:

1. The links failure probabilities are independent.
2. The link failure probabilities are equal at a given time.
3. Similarly, the nodes failure probabilities are equal and independent at a given time.

4. The network has bi-directional links
5. Links and nodes have a performance variable that degrades with time.

The proposed framework is broadly composed by 1) Links and nodes degradation models for reliability evaluation with BM. 2) Monte Carlo method for all-terminal network reliability estimation (for given reliability values of links and nodes). 3) a DNN model trained to learn the reliability values calculated with MC method. 4) Bayesian updating of parameters and network reliability. The four components of the framework are presented in sections 6.3.3.1 to 6.3.3.4 and summarized in section 6.3.3.5.

6.3.3.1. Links and nodes degradation models for reliability evaluation

Consider the actual degradation path of a particular element (link or node) of a network over time is denoted by $D(t), t > 0$. Samples are observed at discrete points in time $t_1, t_2, \dots t_j, \dots$. The observed sample degradation y_{ij} for sample i, at time t_j in a general degradation path model is given as:

$$y_{ij} = D_{ij} + \epsilon_{ij} \tag{108}$$

where $D_{ij} = D(t_{ij}, \beta_{1i}, \dots, \beta_{ki})$ is the actual path of the unit i at time t_{ij} and $\epsilon_{ij} \sim N(0, {\sigma_\epsilon}^2)$ is a residual deviation for the unit i at time t_j. The total number of observations on unit i is m_i. For the i^{th} unit, $\beta_{1i}, \dots, \beta_{ki}$ is a vector of k unknown parameters. A unit i is assumed to fail when its degradation level first reaches to a predefined threshold level D_f.

For simplicity, the unit-to-unit variability in model parameters $\beta_1, \dots, \beta_k$ can be modeled with a multivariate normal distribution with mean vectors $\boldsymbol{\mu_\beta}$ and covariance matrices $\boldsymbol{\Sigma_\beta}$ [14]. It is generally assumed that the random parameters $\beta_1, \dots, \beta_k$ are independent of the ϵ_{ij} and that σ_ϵ is constant. Let $\boldsymbol{\theta_\beta} = (\boldsymbol{\mu_\beta}, \boldsymbol{\Sigma_\beta})$ denote the overall population/process parameters.

The likelihood for the random-parameter degradation model is given as [14]:

$$L(\boldsymbol{\theta_\beta}, \sigma_\epsilon | DATA) = \prod_{i=1}^{n} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \left[\prod_{j=1}^{m_i} \frac{1}{\sigma_\epsilon} \varphi_{\mathrm{nor}}(\zeta_{ij}) \right] \times f_{\boldsymbol{\beta}}(\beta_{1i}, \dots, \beta_{ki}; \boldsymbol{\theta_\beta}) d\beta_{1i}, \dots, \beta_{ki} \quad (109)$$

where $\zeta_{ij} = [y_{ij} - D(t_{ij}, \beta_{1i}, \dots, \beta_{ki})]/\sigma_\epsilon$, and $f_{\boldsymbol{\beta}}(\beta_{1i}, \dots, \beta_{ki}; \boldsymbol{\theta_\beta})$ is the multivariate normal distribution density function. The evaluation of equation (109) requires the numerical approximation of n integrals of dimension k (n is the number of sample paths and k is the number of parameters for each path). Therefore, maximizing equation (109) with respect to $(\boldsymbol{\mu_\beta}, \boldsymbol{\Sigma_\beta}, \sigma_\epsilon)$ directly can be extremely difficult, although there are methods [283] and software packages, e.g., ‘nmle’ [199] to calculate the maximum likelihood estimates (MLE). As an alternative to MLE methods, Bayesian estimation approaches, which allow incorporation of prior information, are receiving more attention recently and will be considered for this study to obtain both initial parameter estimates and updated estimates.

Considering degradation of a performance variable, a fixed value D_f is used to denote critical level for the degradation path. The failure time T is defined as the time when the actual path $D(t)$ crosses the critical degradation level D_f. Therefore, if a unit fails at time t, i.e., the degradation level first reaches D_f at time t, the cumulative distribution function (CDF) of the failure-time distribution is given as:

$$F(t) = \Pr(T \leq t) = F(t, \boldsymbol{\theta_\beta}) = \Pr[D(t, \beta_1, \dots, \beta_k) \geq D_f] \quad (110)$$

For most practical cases, where $D(t)$ is nonlinear and $\beta_1, \dots, \beta_k$ are random parameters, there is not closed-form expression for $F(t)$, and it has to be evaluated by methods such as numerical integration or MC [14].

In the present study, degradation will be considered for both links and nodes. Therefore, there will be a CDF of failure-time distribution for links, $F_L(t)$, and another CDF for nodes, $F_N(t)$. The functions p_L, and p_N, can be calculated by $p_L(t) = 1 - F_L(t)$, and $p_N(t) = 1 -$

$F_N(t)$, respectively. Furthermore, the functions $p_L(t)$ and $p_N(t)$ estimated from degradation data can be used to evaluate the overall network reliability by using the approach described later in section 6.3.3.3.

The degradation patterns (e.g., linear, convex, or concave), and in consequence the reliability functions $p_L(t)$ and $p_N(t)$, will depend on the degradation characteristics of the links and the nodes of a particular network. The purpose of this section is to provide a generic approach to estimate the reliability of a network considering degradation of links and nodes, depending on the data available. To illustrate the detailed application of the proposed framework, particular models are described in the next sections (6.3.3.1.1 and 6.3.3.1.2) for links and nodes, as well as the expressions to evaluate the corresponding reliability functions $p_L(t)$ and $p_N(t)$. Although, sections 6.3.3.1.1 and 6.3.3.1.2 present a nonlinear degradation model to characterize links and a (transformed to) linear degradation model to describe nodes of a network, different degradation profiles can be considered depending on the actual network and its components.

6.3.3.1.1. Link's degradation modeling for reliability evaluation

Usually, links represent communication paths between the nodes in a network. For instance, the links may represent the optical fibers of a network, which can be affected by crack growth [284, 285]. For instance, when the fiber is exposed to sustained stress, degradation occurs as crack growth [286, 287]. Moreover, cracks are a kind of failure mechanisms that lead to degradation of light transmission capabilities [288]. Hence, a crack growth model is assumed for degradation of links in this section.

Let $a(t)$ be the size of a crack at time t. By the Paris-rule model [289], we have:

$$\frac{da(t)}{dt} = C \times [\Delta K(a)]^m \tag{111}$$

where, C and m are material properties and $\Delta K(a)$ is the stress intensity range function. Considering a small crack, $\Delta K(a) = Stress\sqrt{\pi a}$. The solution to the differential equation (111) is:

$$a(t) = \begin{cases} \left[\left(a(0)\right)^{1-\frac{m}{2}} + \left(1 - \frac{m}{2}\right) \times C \times \left(Stress\sqrt{\pi}\right)^m \times t\right]^{\frac{2}{2-m}}, & m \neq 2 \\ a(0) \times \exp\left[C \times \left(Stress\sqrt{\pi}\right)^2 \times t\right] & m = 2 \end{cases} \tag{112}$$

Considering the crack size as the links performance measure, the degradation path is given as $D_{ij} = a(t)$. Let $Stress = 1$, $\beta_1 = C \times \left(\sqrt{\pi}\right)^m$, and $\beta_2 = m$. β_1 and β_2 are modeled by a bivariate normal distribution with parameters $\left(\mu_{\beta_1}, \mu_{\beta_2}, \sigma_{\beta_1}, \sigma_{\beta_2}, \rho\right)$. Therefore, the general degradation path model for the observed degradation is given as:

$$y_{ij} = a\left(t_{ij}, a(0), \beta_{1i}, \beta_{2i}\right) + \epsilon_{ij} \tag{113}$$

To estimate the parameters, we propose to use BM instead of maximizing equation (109) with respect to $\left(\boldsymbol{\mu_\beta}, \boldsymbol{\Sigma_\beta}, \sigma_\epsilon\right)$ or employing software such as ‘nmle’ [199] R package to calculate the MLE estimates. Bayesian estimation is a promising alternative to maximum likelihood (ML) and is getting attention recently. One advantage of Bayesian estimation is that modern Markov Chain Monte Carlo (MCMC) methods with low-information prior distributions provide estimation results close to ML estimates [290]. In addition, BM provides not only point estimates but also credible intervals, which can be derived from MCMC draws [290]. Credible intervals are commonly used to describe the Bayesian analog to non-Bayesian confidence intervals [290]. Another benefit of BM is that prior information, if available, can be incorporated into the analysis [291, 292], providing improvements in precision or testing cost savings. Moreover, from a practical point of view, Bayesian methods can handle complicated data-model combinations for which there is no maximum likelihood (ML) software or for which implementing ML would be extremely challenging. For these reasons, in this study, Bayesian estimation is used for both initial estimation of parameters and updating of parameters with new degradation data. Bayesian estimation will be described in section 6.3.3.1.3.

Once the parameters β_1 and β_2 are estimated, an expression is needed for the links reliability $p_L(t)$, as a function of time. Since the parameters β_1 and β_2 follow a bivariate normal distribution with parameters $\mu_{\beta_1}, \mu_{\beta_2}, \sigma^2_{\beta_1}, \sigma^2_{\beta_2}$ and ρ, then a numerical integration approach can be given as:

$$F_L(t) = \int_{-\infty}^{\infty} \Phi_{\text{nor}}\left[-\frac{g(D_f, t, \beta_1) - \mu_{\beta_2|\beta_1}}{\sigma_{\beta_2|\beta_1}}\right]\frac{1}{\sigma_{\beta_1}}\,\varphi_{\text{nor}}\left(\frac{\beta_1 - \mu_{\beta_1}}{\sigma_{\beta_1}}\right)d\beta_1 \qquad (114)$$

where, $\Phi_{\text{nor}}(\cdot)$ is the standardized normal CDF, $\varphi_{\text{nor}}(\cdot)$ is the standardized normal PDF, $g(D_f, t, \beta_1)$ is the value of β_2 that gives $D(t) = D_f$ for specified β_1, $\mu_{\beta_2|\beta_1} = \mu_{\beta_2} + \rho\sigma_{\beta_2}\left(\frac{\beta_1 - \mu_{\beta_1}}{\sigma_{\beta_1}}\right)$, and $\sigma^2_{\beta_2|\beta_1} = \sigma^2_{\beta_2}(1-\rho^2)$

Therefore, the links reliability is given as:

$$p_L(t) = 1 - \int_{-\infty}^{\infty} \Phi_{\text{nor}}\left[-\frac{g(D_f, t, \beta_1) - \mu_{\beta_2|\beta_1}}{\sigma_{\beta_2|\beta_1}}\right]\frac{1}{\sigma_{\beta_1}}\,\varphi_{\text{nor}}\left(\frac{\beta_1 - \mu_{\beta_1}}{\sigma_{\beta_1}}\right)d\beta_1 \qquad (115)$$

6.3.3.1.2. Node's degradation modeling for reliability evaluation

Degradation may also affect the transmitter nodes in a fiber optic network [285, 293]. Thus, a light emitting diode (LED) degradation model is considered for nodes, as LEDs are used to generate the light in fiber optic networks [294].

In this section we consider the modeling for data from accelerated degradation test on LEDs reported by Pascual et al. [282]. Sample LEDs were tested at six different combinations of junction temperature and current. The performance characteristic was the light output. An approximately linear degradation path is obtained by applying a square-root transformation for time axis only. Standard acceleration models are applied for temperature and current [14, 190, 206]. The Arrhenius transformation is used on junction temperature in degrees Celsius (equation (116)) and the Black's law for current (in milliamps) acceleration (equation (117)).

$$x_1 = \frac{11605}{T_C + 273.15} \tag{116}$$

$$x_2 = \log(I_{mA}) \tag{117}$$

The mixed-effects model for the actual LED light-output degradation for the sample i at (transformed) time τ_j, for test condition k, based on normalized data is given as [290]:

$$D_{ijk,N} = 1 + \beta_{1,N}(x_{1k} - x_1^0)\tau_j + \beta_{2,N}(x_{2k} - x_2^0)\tau_j + b_i\tau_j \tag{118}$$

where $\tau \propto \sqrt{t}$ because of the square-root transformation for time. The values x_1^0 and x_2^0 should be chosen to be near the center of the respective transformed variables to improve numerical stability [290]. The subscripts "$, N$" are used to refer to node's degradation path and parameters, to avoid confusion with the degradation path and parameters of links.

To model the randomness in the slopes for the different LED samples, b_i is described by a normal distribution: $b_i \sim N(\mu_b, {\sigma_b}^2)$. The model for the observed degradation is then given as:

$$Y_{ijk,N} = D_{ijk,N} + \varepsilon_{ijk,N} = 1 + \beta_{1,N}(x_{1k} - x_1^0)\tau_j + \beta_{2,N}(x_{2k} - x_2^0)\tau_j + b_i\tau_j + \varepsilon_{ijk,N} \tag{119}$$

where $\varepsilon_{ijk,N} \sim N(0, {\sigma_\varepsilon}^2)$ describes the measurement error, with the independence assumption of b_i and across time [290].

The parameters $\beta_{1,N}$, $\beta_{2,N}$ μ_b, and σ_b will be estimated by BM as well, as discussed earlier. The estimated parameters together with the critical light-output level $D_{f,N}$ will determine the expression for the nodes reliability $p_N(t)$, as a function of time.

Since the light-output exhibits a decreasing degradation pattern, the probability of failure is given as:

$$F_N(t) = \Pr(T \leq t) = \Pr\left[D \leq D_{f,N}\right] \tag{120}$$

$$F_N(t) = \Pr\left[1 + \beta_{1,N}(x_1 - x_1^0)\tau + \beta_{2,N}(x_2 - x_2^0)\tau + b_i\tau \leq D_{f,N}\right] \tag{121}$$

$$F_N(t) = \Pr\left[b_i \leq \frac{D_{f,N} - \left(1 + \beta_{1,N}(x_1 - x_1^0)\tau + \beta_{2,N}(x_2 - x_2^0)\tau\right)}{\tau}\right] \tag{122}$$

Since $b_i \sim N(\mu_b, {\sigma_b}^2)$,

$$F_N(t) = \Phi_{\text{nor}}\left[\frac{\frac{D_{f,N} - \left(1 + \beta_{1,N}(x_1 - x_1^0)\tau_j + \beta_{2,N}(x_2 - x_2^0)\tau_j\right)}{\tau} - \mu_b}{\sigma_b}\right] \tag{123}$$

$$F_N(t) = \Phi_{\text{nor}}\left[\frac{D_{f,N} - \left(1 + \beta_{1,N}(x_1 - x_1^0)\tau + \beta_{2,N}(x_2 - x_2^0)\tau + \mu_b\tau\right)}{\tau\sigma_b}\right] \tag{124}$$

Let $\mu = 1 + \beta_{1,N}(x_1 - x_1^0)\tau + \beta_{2,N}(x_2 - x_2^0)\tau + \mu_b\tau$, then

$$F_N(t) = \Phi_{\text{nor}}\left[\frac{D_{f,N} - \mu}{\tau\sigma_b}\right] \tag{125}$$

Therefore, the nodes reliability is given as:

$$p_N(t) = 1 - \Phi_{\text{nor}}\left[\frac{D_{f,N} - \mu}{\tau\sigma_b}\right] \tag{126}$$

6.3.3.1.3. Bayesian approach for parameter estimation and reliability evaluation of links and nodes

Bayesian approach is based on Bayes' theorem, which relates different kinds of conditional probabilities (or conditional probability density functions) to one another. The Bayesian method for statistical inference provides a mechanism to combine available data with prior information to obtain a posterior distribution that can be used to make inferences about some vector $\boldsymbol{\theta}$ of unknown parameters. Bayes' theorem for continuous parameters in $\boldsymbol{\theta}$ is given as:

$$f(\boldsymbol{\theta}|\mathrm{DATA}) = \frac{L(\mathrm{DATA}|\boldsymbol{\theta})f(\boldsymbol{\theta})}{\int L(\mathrm{DATA}|\boldsymbol{\theta})f(\boldsymbol{\theta})d\boldsymbol{\theta}} \tag{127}$$

where the joint prior distribution $f(\boldsymbol{\theta})$ provides the available prior information about the unknown parameters in $\boldsymbol{\theta}$. $f(\boldsymbol{\theta}|\mathrm{DATA})$ is the joint posterior distribution for $\boldsymbol{\theta}$, which combines the information from the data and the prior distribution. $L(\mathrm{DATA}|\boldsymbol{\theta})$ is the likelihood function and depends on the assumed model for the data and the data itself. This function must be proportional to the probability of the data. The denominator of the equation (127) is a normalizing constant that assures that the joint distribution is a proper probability distribution [290].

One of the reasons for controversy on the use of Bayesian methods is that it is possible that the prior distribution will have a strong influence on the resulting inferences, especially when the amount of information from the data is scarce. Therefore, the joint prior distribution must be carefully specified. If there is no agreement among interested parties, e.g., manufacturers and customers, a convenient alternative is to use noninformative prior distributions. When the joint prior distribution is diffuse or relatively flat over the values of $\boldsymbol{\theta}$ for which the likelihood is non-negligible, and if the data dominates the joint prior, the likelihood is approximately proportional to the joint posterior distribution. As a result, Bayesian inferences are similar to those obtained from non-Bayesian methods, e.g., ML [290]. In this study, Bayesian inference with low-information distributions will be employed for initial estimation of parameters.

The computation of the joint posterior distribution for $\boldsymbol{\theta}$ (equation (127)) in closed form is impossible in many cases, because computing the integral in the denominator can be intractable. As an alternative, modern methods for Bayesian analysis overcome this complexity by obtaining inferences based on draws from the joint posterior distribution [290]. A powerful method for simulating a sample from a particular joint posterior distribution is Markov chain

Monte Carlo (MCMC) [292]. Gibbs sampling and MCMC [292] provides an efficient method to simulate draws from a discrete time continuous-space Markov chain. After reaching a steady-state, the sequence of draws provides a sample from the desired joint posterior distribution [290]. The MCMC simulation proposed is summarized in the following Algorithm (6.2). WinBUGS software is an excellent alternative to perform the MCMC simulation [295] and will be used in the case study analysis to estimate the posterior parameters.

Algorithm 6.2

Step 1: Set low-information prior distributions for parameters of the distributions assumed for random parameters. For example, set low-information prior distributions for $\mu_{\beta_1}, \mu_{\beta_2}, \sigma^2_{\beta_1}, \sigma^2_{\beta_2}$ and ρ, for the links degradation model

Step 2: Generate a large number (T) of MCMC sample draws using prior distributions and degradation data from the assumed distributions until equilibrium is reached

Step 3: Cut off (“burn-in”) the first B (e.g., B = 4,000) number of initial draws to omit the noise effect

Step 4: Monitor the convergence of posterior equilibrium, if not, generate more sample draws.

Step 5: Use MCMC sample draws of the model parameters (for both links and nodes degradation models) with equations (115) and (126), to evaluate the links and nodes reliability, respectively, for a large number of time values (e.g., between 0 to 10,000 hours).

Step 6: From the reliability values evaluated for each time value, obtain the point estimates (median values) and credible intervals (e.g., 95% credible intervals).

6.3.3.2. Monte Carlo method for all-terminal network reliability estimation

We use the Monte Carlo method (Algorithm 6.1.) proposed in section 6.2.2.1.

6.3.3.3. Deep neural network model (DNN model)

Given a network $G(N, L, p_L, p_N)$, with links reliability p_L, and nodes reliability p_N, the all-terminal network reliability can be estimated by equation (105), with $R_{MC_{links}}$ evaluated by Algorithm 6.1. To speed-up the calculation, a DNN is proposed to be trained with the estimated reliability of links $R_{MC_{links}}$ as the target, as a function of the links reliability value p_L as the input. The reliability of links for a set of links reliability values (p_L) can be calculated by using the MC proposed method, before the training of the DNN. The DNN (to estimate the reliability of links R_{links}) along with equation (103) (to calculate the reliability of nodes R_{nodes}) conform a DNN model. The DNN model is expected to predict the network reliability (for new given p_L and p_N) accurately and quickly. A representation of the DNN model is shown in Figure 44.

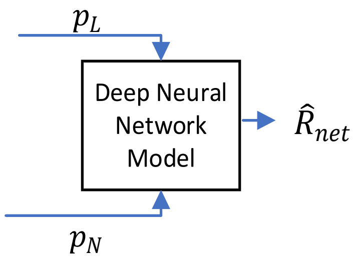


Figure 44. Deep neural network to estimate the reliability of a network

A base dataset is formed by the pairs (X_t, y_t), where X_t is the links reliability value (p_L) and y_t is corresponding estimated reliability of links $(R_{MC_{links}})$, for each element in a set of link reliability values, e.g., {0.01, 0.02, …, 0.99, 1.00}.

6.3.3.3.1. DNN architecture

The proposed DNN architecture is based on feed-forward neural networks as they have proven to be effective function approximators [266]. Furthermore, a fully connected multi-layer perceptron (MLP) structure is employed. A sigmoid activation function is used at the output layer to ensure the reliability predicted falls within the range [0,1], a feature that some previous ANN-based works lack [42, 45].

In addition, a dropout layer is placed before the output layer to avoid overfitting by randomly dropping neurons during the training [252]. To summarize, the architecture is a stack of the following layers: input: (X_t), fully connected hidden layers, dropout, output with sigmoid activation: $\widehat{y_t}$.

6.3.3.3.2. Training and evaluating

Once a base dataset is available and the architecture of the DNN is defined, the remaining phases are training and evaluating. Five-fold cross-validation is considered to compare the results obtained with different test sets from the same base dataset. The base dataset is (randomly) divided into five subsets of the same size. DNN training uses all but one subset, and the excluded subset is considered a test set for the trained DNN. After training, the DNNs are evaluated in terms of the error using the testing datasets from cross-validation.

6.3.3.3.3. Calculating the all-terminal network reliability

The cross-validation error is used to select the best DNN. The best trained DNN can be used to estimate the reliability of links ($\hat{R}_{links}$) for a given links reliability value (p_L). The reliability of links R_{links} depends not only on p_L but also on the topology of the network. These dependences are incorporated by the MC Algorithm 6.1 and learned by the DNN. Moreover, as expected, such dependences (captured by $\hat{R}_{links}$) are reflected in the estimated network reliability $\hat{R}_{net}$, as shown in equation (106). Further, the reliability of nodes (R_{nodes}) for a given nodes reliability value (p_N) is given by equation (103). Therefore, the estimated all-terminal reliability of the network is given by the proposed DNN model as:

$$\hat{R}_{net} = \underbrace{DNN(p_L)}_{\hat{R}_{links}} \times R_{nodes}(p_N) = DNN(p_L) \times p_N^{|N|} \qquad (128)$$

where $DNN(p_L)$ represents the estimation of the reliability of links provided by the best DNN, i.e., $\hat{R}_{links}$

6.3.3.4. Bayesian updates of parameters and network reliability prediction

To further reduce the uncertainty in parameter estimates and network reliability prediction obtained from initial (possibly ADT) data and initial Bayesian parameter estimation, the framework allows to update the initial estimations with new data. The new data is incorporated according to the algorithm 6.3 proposed below:

Algorithm 6.3

Step 1: From the initial MCMC draws obtained by using Algorithm 6.2, obtain informative prior distributions for parameters of the distributions assumed for random parameters. For example, set low-information prior distributions for $\mu_{\beta_1}, \mu_{\beta_2}, \sigma^2_{\beta_1}, \sigma^2_{\beta_2}$ and ρ in the case of the links degradation model. Besides information from historical data, experts opinion is another source of prior information [290].

Step 2: Generate a large number (T_2) of MCMC sample draws using prior distributions and new degradation data from the assumed distributions and until equilibrium is reached.

Step 3: Cut off ("burn-in") the first B_2 (e.g., B_2 = 4,000) number of initial draws to omit the noise effect.

Step 4: Monitor the convergence of posterior equilibrium, if not, generate more sample draws.

Step 5: Use MCMC sample draws of the model parameters (for both links and nodes degradation models) with equations (115) and (126) to evaluate the updated links and nodes reliability, respectively, for a large number of time values (e.g., between 0 to 10,000 hours).

Step 6: From the updated reliability values evaluated for each time value, obtain the updated point estimates (median values) and credible intervals (e.g., 95% credible intervals).

This updating process can be repeated subsequently, if newer data become available as needed. Therefore, the proposed framework provides a method for continuous updating of parameters and reliability predictions of links, nodes, and network.

6.3.3.5. Framework for network reliability estimation and updating of parameters

To summarize, the overall proposed framework can estimate the reliability of a network, considering degradation data of links and nodes. The framework is broadly composed by a links degradation model, a nodes degradation model, and a DNN model (equation (106) and Figure 44) trained using reliability values obtained by the Monte Carlo Algorithm 6.1. The links degradation model (Figure 45) provides the links reliability $p_L(t)$ based on degradation data. This model considers the updating of parameters if new data become available. Similar model is considered for the nodes degradation model, with the degradation data generated from nodes and an appropriate degradation model. The overall proposed framework is represented in Figure 46. The outputs $p_L(t)$ and $p_N(t)$ are fed to the DNN model for it to predict the network reliability $\hat{R}_{net}$. Moreover, the nodes reliability, the links reliability, and the overall network reliability can be updated as new data are available.

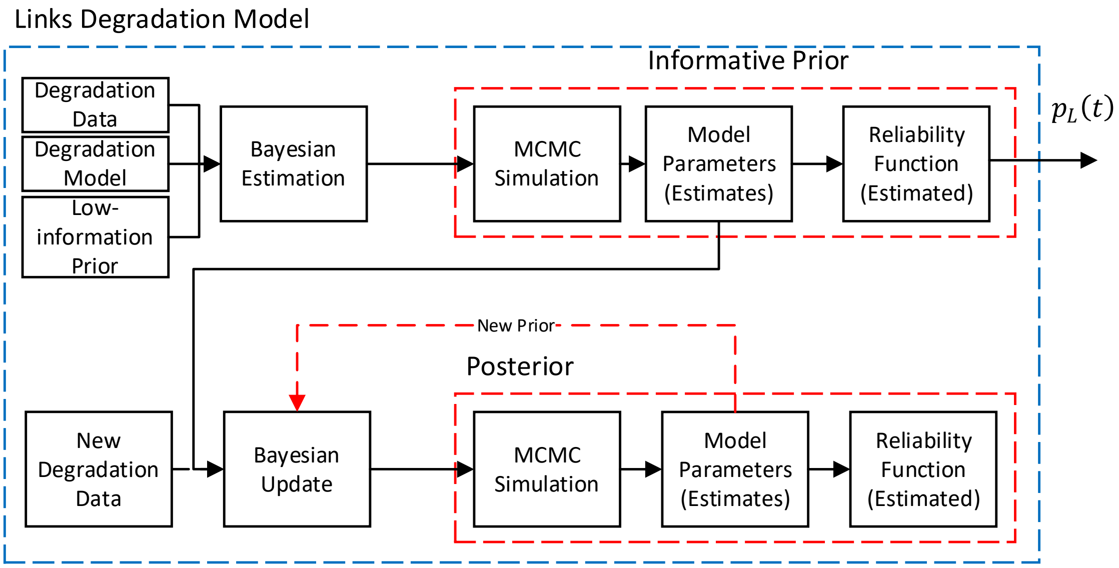


Figure 45. Links degradation model

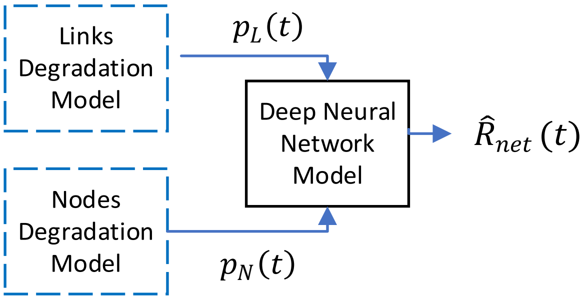


Figure 46. Framework for all-terminal network reliability estimation

6.3.4. Case study

To demonstrate the proposed framework, the real-world network topology Ion (125 nodes, 150 links, New York, USA [269, 281]), shown in Figure 47, was analyzed using the proposed approach (Figure 47). Real degradation data was considered to simulate additional degradation data for links [59]. Similarly, real degradation data was obtained for nodes from [282]. Since a crack growth model was assumed for the links, degradation data for 150 sample units would be needed. Lu and Meeker [59] reported crack size degradation data for 21 sample units only. In this study, the initial degradation parameters were estimated from such available real degradation data. Based on these estimated parameters, degradation data for 150 sample units were simulated. On the other hand, a LED light-output degradation model was assumed for the nodes and hence, degradation data for 125 sample units are required. Pascual et al. [282] provided appropriate LED light-output degradation data. They reported degradation data for six groups, with 30 sample units per group and each group corresponding to a different test condition. Since the data from one of the extreme test conditions were reported to cause serious model-fit problems [282], possibly due to the occurrence of new failure mechanisms, we decided not to consider that set of data in this study. Also, data for five sample units from each of the remaining five groups were removed to obtain degradation data for a total of 125 sample units (five groups with 25 sample units per group).

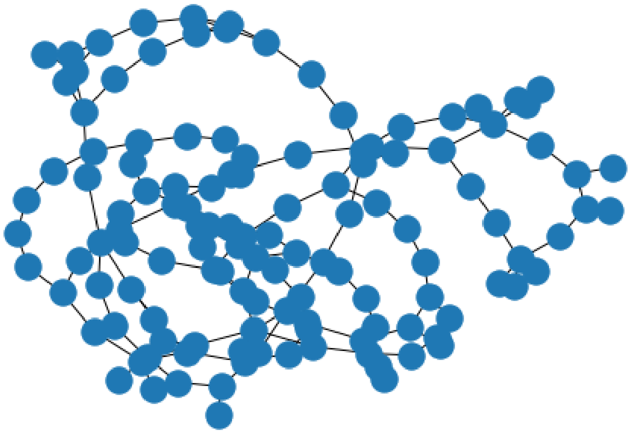


Figure 47. ION network graphical representation

6.3.4.1. Links degradation modeling for reliability evaluation

For degradation modeling of links, the original degradation data for 21 sample units obtained from [59] were considered (shown in Figure 48). Considering the degradation model given by equation (113), the random parameters β_1 and β_2 are modeled by a bivariate normal distribution with parameters $(\mu_{\beta_1}, \mu_{\beta_2}, \sigma_{\beta_1}, \sigma_{\beta_2}, \rho)$, and the residual deviation ϵ is described by a normal distribution with mean zero and standard deviation σ_ϵ. We propose Bayesian estimation of such parameters. These parameters will be used to carryout simulation and obtain degradation data for 150 links of the network considered in this study. In addition to degradation data, prior distributions are needed for the parameters $\mu_{\beta_1}, \mu_{\beta_2}, \sigma_{\beta_1}, \sigma_{\beta_2}, \rho$, and σ_ϵ.

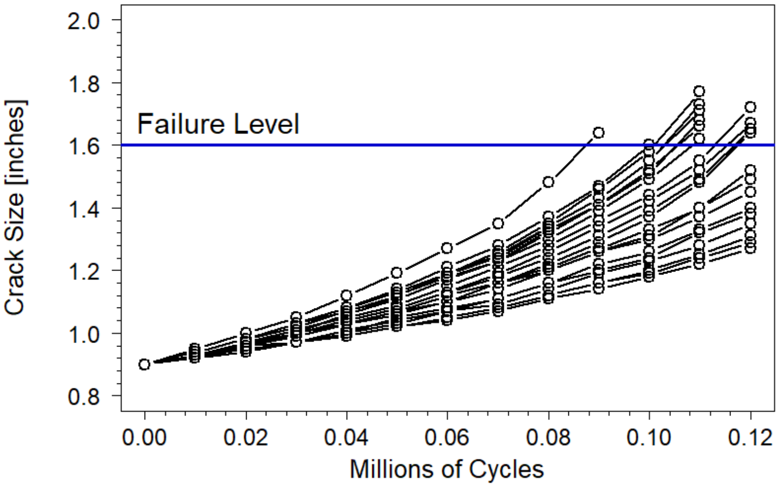


Figure 48. Crack growth data

The vector of random unknown parameters of the path model $\begin{pmatrix}\beta_1\\ \beta_2\end{pmatrix}$ described by a bivariate normal distribution, can be modeled as:

$$\begin{pmatrix}\beta_1\\ \beta_2\end{pmatrix} \sim N\left(\begin{pmatrix}\mu_{\beta_1}\\ \mu_{\beta_2}\end{pmatrix}, \begin{pmatrix}{\sigma_{\beta_1}}^2 & \rho\sigma_{\beta_1}\sigma_{\beta_2}\\ \rho\sigma_{\beta_1}\sigma_{\beta_2} & {\sigma_{\beta_2}}^2\end{pmatrix}\right) \tag{129}$$

Or equivalently:

$$\begin{pmatrix}\beta_1\\ \beta_2\end{pmatrix} \sim N(\boldsymbol{\mu_\beta}, \boldsymbol{\Sigma_\beta}) \tag{130}$$

where, $\boldsymbol{\mu_\beta} = \begin{pmatrix}\mu_{\beta_1}\\ \mu_{\beta_2}\end{pmatrix}$, and $\boldsymbol{\Sigma_\beta} = \begin{pmatrix}{\sigma_{\beta_1}}^2 & \rho\sigma_{\beta_1}\sigma_{\beta_2}\\ \rho\sigma_{\beta_1}\sigma_{\beta_2} & {\sigma_{\beta_2}}^2\end{pmatrix}$

Then, in the next paragraphs, low-information prior distributions are considered for the parameters $\boldsymbol{\mu_\beta}$, $\boldsymbol{\Sigma_\beta}$, and σ_ϵ.

For $\boldsymbol{\mu_\beta}$ we assume a low-information bivariate normal distribution centered in $\mathbf{0}$ with no correlation between and large variances for μ_{β_1} and μ_{β_2} [296]. In WinBUGS, the multivariate normal distribution is specified in terms of a mean vector and a precision matrix (inverse of covariance matrix), as shown in equation (131):

$$\boldsymbol{\mu_\beta} = \binom{\mu_{\beta_1}}{\mu_{\beta_2}} \sim N\left(\boldsymbol{\mu_{\mu_\beta}}, \mathbf{T_{\mu_\beta}}^{-1}\right) \tag{131}$$

where $\mathbf{T_{\mu_\beta}}$ is the precision matrix and given as $\mathbf{T_{\mu_\beta}} = \boldsymbol{\Sigma_{\mu_\beta}}^{-1}$

On the other hand, to represent a vague prior knowledge for $\boldsymbol{\Sigma_\beta}$, a low-information Wishart distribution [291, 297] is used to describe the precision matrix $\mathbf{T_\beta}$, where $\mathbf{T_\beta} = \boldsymbol{\Sigma_\beta}^{-1}$, as shown in equation (132).

$$\mathbf{T_\beta} = \boldsymbol{\Sigma_\beta}^{-1} \sim W_p\left(\mathbf{S_0}^{-\mathbf{1}}, \nu_0\right) \tag{132}$$

The parameters of a Wishart distribution W_p of a $p \times p$ symmetric positive definite matrix are the degrees of freedom ν_0, and the $p \times p$ positive definite scale matrix $\mathbf{S_0}^{-\mathbf{1}}$. In WinBUGS the inverse of the scale matrix, i.e., $\mathbf{S_0}$, must be specified.

In this case study $\mathbf{T_{\mu_\beta}}^{-1}$ is a 2×2 matrix, then $p = 2$. To represent low prior knowledge, the (low-information) Wishart distribution has the degrees of freedom as small as possible [291, 296], i.e., $\nu_0 = p$, and $\mathbf{S_0}$ represents a prior guess at the order of magnitude of the covariance matrix $\boldsymbol{\Sigma_\beta}$ [296].

Finally, a prior distribution needs to be defined for the parameter σ_ϵ, which is considered to describe the residual deviation as $\epsilon_{ij} \sim N(0, {\sigma_\epsilon}^2)$. In WinBUGS, a precision (inverse of variance) parameter is used to specify normal distributions. Therefore, using WinBUGS parameters, the residual deviation can be described as:

$$\epsilon_{ij} \sim N(0, {\tau_\epsilon}^{-1}) \tag{133}$$

where $\tau_\epsilon = {\sigma_\epsilon}^{-2}$

τ_ϵ can be described by a gamma distribution $Gamma(\alpha, \beta)$ with shape and rate parameters α, β, respectively:

$$\tau_\epsilon = {\sigma_\epsilon}^{-2} \sim Gamma(\alpha, \beta) \tag{134}$$

A common low-information prior Gamma distribution is obtained by letting $\alpha = \beta = 0.001$ [296], which provides a mean of 1 and a large variance of 1000.

Table 21 summarizes the low-information prior distributions used for initial estimation of parameters.

Table 21. Low-information prior distribution specifications for links degradation model

Parameter	Prior distribution
$\boldsymbol{\mu_\beta} = \begin{pmatrix} \mu_{\beta_1} \\ \mu_{\beta_2} \end{pmatrix}$	$N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1.0 \times 10^{-6} & 0 \\ 0 & 1.0 \times 10^{-6} \end{pmatrix}^{-1}\right)$
${\boldsymbol{\Sigma}_\boldsymbol{\beta}}^{-1} = \mathbf{T}_\boldsymbol{\beta} = \begin{pmatrix} {\sigma_{\beta_1}}^2 & \rho\sigma_{\beta_1}\sigma_{\beta_2} \\ \rho\sigma_{\beta_1}\sigma_{\beta_2} & {\sigma_{\beta_2}}^2 \end{pmatrix}^{-1}$	$W_2\left(\begin{pmatrix} 1.0 \times 10^{-3} & 0 \\ 0 & 1.0 \times 10^{-3} \end{pmatrix}^{-1}, 2\right)$
${\sigma_\epsilon}^{-2} = \tau_\epsilon$	$Gamma(0.001{,}0.001)$

To summarize, a WinBUGS model was built considering the original degradation data [59], the degradation model given by equation (113), the distributions assumed for the parameters β_1, β_2 and ϵ, and the low-information prior distributions assumed for parameters $\boldsymbol{\mu_\beta}$, $\boldsymbol{\Sigma_\beta}$, and σ_ϵ (see Table 21). The initial 4,000 MCMC sample draws were dropped (“burn-in”) and the sample draws were “thinned” [295] to reduce autocorrelation, by setting a lag parameter L of 30, i.e.; in the sequence, every 30th value was retained. In general, L would be larger if autocorrelation is stronger in the preliminary experiments [290]. The point estimates are obtained by the median values from the MCMC sample draws [298]. The results are: $\boldsymbol{\hat{\mu}_\beta} = \begin{pmatrix} 3.717 \\ 5.225 \end{pmatrix}$,

$\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}} = \begin{pmatrix} 0.5219 & -0.1735 \\ -0.1735 & 0.2349 \end{pmatrix}$, and $\hat{\sigma}_{\epsilon} = 0.008008$. WinBUGS also provides kernel density estimations. As an example, Figure 49 shows the kernel density estimations of parameter $\boldsymbol{\mu}_{\boldsymbol{\beta}} = \begin{pmatrix} \mu_{\beta_1} \\ \mu_{\beta_2} \end{pmatrix}$.

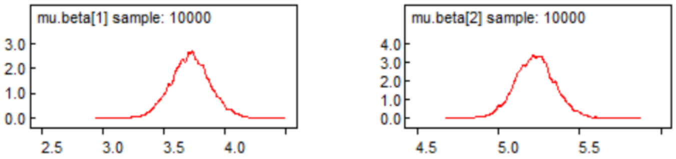


Figure 49. Bayesian kernel density estimation of parameter $\boldsymbol{\mu}_{\boldsymbol{\beta}}$

Using the estimated parameters $\widehat{\boldsymbol{\mu}}_{\boldsymbol{\beta}}$, $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}$, and $\hat{\sigma}_{\epsilon}$,degradation data were simulated for 150 sample units to match the number of links of the network analyzed. 12 measures were simulated for each sample unit, i.e., considering readings at 0.01, 0.02, 0.03, …, 0.12 Mcycles. The simulated data were divided in two parts: “initial data” , i.e., readings at times from 0.01 to 0.08 Mcycles and “new data”, i.e., readings at times from 0.09 to 0.12 Mcycles. The purpose of this division is to illustrate the initial Bayesian estimation of parameters with the “initial data” and subsequently the Bayesian updating of parameters as “new data” become available.

6.3.4.1.1. Bayesian estimation of parameters

A WinBUGS model was built considering the “initial data”, the degradation model given by equation (113), the distributions assumed for the parameters β_1, β_2 and ϵ, and the low-information prior distributions assumed for parameters $\boldsymbol{\mu}_{\boldsymbol{\beta}}$, $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$, and σ_{ϵ} as given in Table 21. The initial 4,000 MCMC sample draws were dropped (“burn-in”) and the sample draws were “thinned” to reduce autocorrelation, by setting a lag parameter L of 60. The point estimates obtained by the median values from the MCMC sample draws are: $\widehat{\boldsymbol{\mu}}_{\boldsymbol{\beta}} = \begin{pmatrix} 3.608 \\ 5.370 \end{pmatrix}$, $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}} =$

$\begin{pmatrix}0.4458 & -0.2215\\ -0.2215 & 0.2604\end{pmatrix}$, and $\hat{\sigma}_{\epsilon} = 0.008012$. Figure 50 shows the kernel density estimations of parameter $\boldsymbol{\mu_\beta} = \begin{pmatrix}\mu_{\beta_1}\\ \mu_{\beta_2}\end{pmatrix}$.

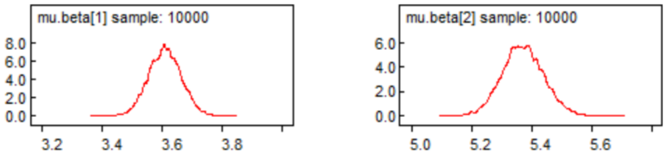


Figure 50. Bayesian kernel density estimation of parameter $\boldsymbol{\mu_\beta}$ for “initial data”

6.3.4.1.2. Links reliability estimation

Both “initial data” and “new data”, for links, have Mcycles as “time” axis, whereas nodes degradation data, considered in section 6.3.4.2 have hours in the time axis. Therefore, to make the time units consistent for reliability calculations, an arbitrary scaling was adopted by setting 64,000 hours as equivalent to one Mcycle for links “initial data” and “new data”.

The 10,000 MCMC sample draws from the joint posterior distributions of the model parameters were used in equation (115) to obtain links reliability draws for 1,001 time values between 0 and 10,000 hours. The median values (solid line) as well as the 95% credible bounds (dashed lines) are shown in Figure 51.

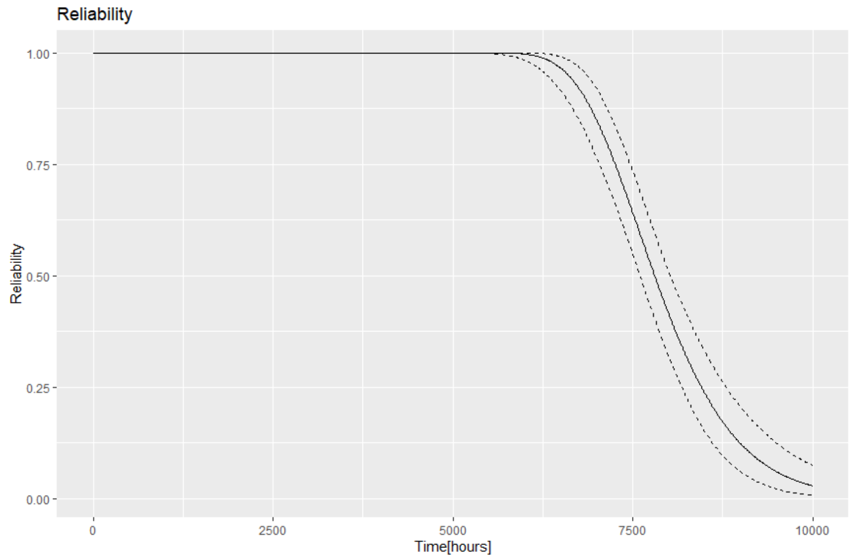


Figure 51. Links reliability and 95% credible intervals

6.3.4.1.3. Bayesian updating of parameters

As a result of the proposed time scaling, “new data” include degradation measures at time = 5,760 hours (i.e., 64,000 hours/Mcycle × 0.09 Mcycles), 6,400 hours, 7,040 hours, and 7,680 hours. A WinBUGS model was built considering the “new data”, the degradation model given by equation (113), the distributions assumed for the parameters β_1, β_2 and ϵ, and informative prior distributions assumed for parameters $\boldsymbol{\mu_\beta}$, and $\boldsymbol{\Sigma_\beta}$. Initial MCMC sample draws obtained in section 6.3.4.1.1 are used to estimate the parameters of informative prior distributions (see Table 22). The prior distribution for $\boldsymbol{\mu_\beta}$ is specified with the MLE estimates obtained for a bivariate normal distribution from the MCMC draws of $\boldsymbol{\mu_\beta}$ (obtained in section 6.3.4.1.1). On the other hand, the prior Wishart distribution for $\boldsymbol{\Sigma_\beta}^{-1}$ is specified by considering that the true covariance matrix $\boldsymbol{\Sigma_0}$ can be estimated by the median values from the MCMC sample draws [298] obtained in section 6.3.4.1.1. To make $\boldsymbol{\Sigma_\beta}$ closely centered around $\boldsymbol{\Sigma_0}$, a large ν_0 is selected [291], whereas $\mathbf{S_0}$ is given by equation (135) [291]:

$$\mathbf{S_0} = (\nu_0 - p - 1)\boldsymbol{\Sigma_0} \tag{135}$$

Since this updating process is intended to take place with “new data” obtained during normal operation, which does not necessarily offer the same testing conditions as for “initial data”, a low-information prior is still considered for the precision parameter ${\sigma_\epsilon}^{-2}$.

The initial 4,000 MCMC sample draws were dropped (“burn-in”) and the sample draws were “thinned” to reduce autocorrelation, by setting a lag parameter L of 100. The point estimates obtained by the median values from the MCMC sample draws are: $\hat{\boldsymbol{\mu}}_{\boldsymbol{\beta}} = \begin{pmatrix} 3.609 \\ 5.333 \end{pmatrix}$, $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}} = \begin{pmatrix} 0.4354 & -0.1479 \\ -0.1479 & 0.2221 \end{pmatrix}$, and $\hat{\sigma}_\epsilon = 0.008611$. Figure 52 shows the kernel density estimations of parameter $\boldsymbol{\mu}_{\boldsymbol{\beta}} = \begin{pmatrix} \mu_{\beta_1} \\ \mu_{\beta_2} \end{pmatrix}$.

Table 22. Informative prior distribution specifications for links degradation model

Parameter	Prior distribution
$\boldsymbol{\mu}_{\boldsymbol{\beta}} = \begin{pmatrix} \mu_{\beta_1} \\ \mu_{\beta_2} \end{pmatrix}$	$N\left(\begin{pmatrix} 3.6084 \\ 5.3716 \end{pmatrix}, \begin{pmatrix} 409.8462 & 131.6914 \\ 131.6914 & 247.8900 \end{pmatrix}^{-1}\right)$
${\boldsymbol{\Sigma}_{\boldsymbol{\beta}}}^{-1} = \mathbf{T}_{\boldsymbol{\beta}} = \begin{pmatrix} {\sigma_{\beta_1}}^2 & \rho\sigma_{\beta_1}\sigma_{\beta_2} \\ \rho\sigma_{\beta_1}\sigma_{\beta_2} & {\sigma_{\beta_2}}^2 \end{pmatrix}^{-1}$	$W_2\left(\begin{pmatrix} 3.1206 & -1.5505 \\ -1.5505 & 1.8228 \end{pmatrix}^{-1}, 10\right)$
${\sigma_\epsilon}^{-2} = \tau_\epsilon$	$Gamma(0.001{,}0.001)$

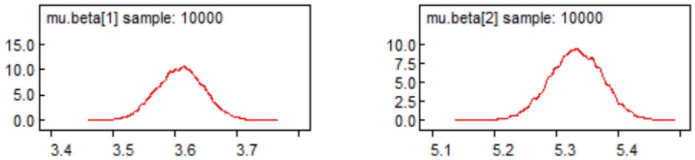


Figure 52. Bayesian kernel density estimation of parameter $\boldsymbol{\mu}_{\boldsymbol{\beta}}$ for “new data”

To verify the robustness of the posterior distributions, we performed sensitivity analysis by considering an alternative plausible model with changes in informative prior distributions [295]. A natural sensitivity analysis is to consider longer-tailed alternatives instead of normal distributions [292]. Hence, in the alternative model, a prior distribution for $\boldsymbol{\mu}_{\boldsymbol{\beta}}$ was specified with the MLE estimates [299] obtained for a multivariate Student’s t distribution from the MCMC draws obtained in section 6.3.4.1.1. In WinBUGS, the (noncentral) multivariate Student's t distribution is specified in terms of mean vector, precision matrix (inverse of covariance matrix), and degrees of freedom. In the alternative model, the prior distribution for $\boldsymbol{\mu_\beta}$ is given as $t\left(\begin{pmatrix}3.6085\\5.3715\end{pmatrix},\begin{pmatrix}422.0977 & 135.4642\\135.4642 & 255.4619\end{pmatrix}^{-1},67\right)$. Table 23 shows the sensitivity of posterior inference in terms of the median and 95% credible intervals obtained from the MCMC sample draws. Minor differences are observed between the posteriors resulted by the model with normal distribution and the alternative model with t distribution. Therefore, the original model that considers the informative prior distributions in Table 22 was used for Bayesian updating of parameters.

Table 23. Posterior median and 95% credible intervals of parameters under different prior assumptions

Parameter	Distribution for $\boldsymbol{\mu_\beta}$			
	Multivariate normal		Multivariate Student's t	
	Posterior median	95% posterior credible interval	Posterior median	95% posterior credible interval
$\boldsymbol{\mu_\beta}$	$\begin{pmatrix}3.609\\5.333\end{pmatrix}$	$[3.533, 3.683]$ $[5.247, 5.416]$	$\begin{pmatrix}3.609\\5.333\end{pmatrix}$	$[3.534, 3.684]$ $[5.248, 5.418]$
$\boldsymbol{\Sigma_\beta}$	$\begin{pmatrix}0.4354 & -0.1479\\-0.1479 & 0.2221\end{pmatrix}$	$\begin{bmatrix}0.3531,\\0.5509\end{bmatrix}$ $\begin{bmatrix}-0.2288,\\-0.0815\end{bmatrix}$ $\begin{bmatrix}-0.2288,\\-0.0815\end{bmatrix}$ $\begin{bmatrix}0.1604,\\0.3093\end{bmatrix}$	$\begin{pmatrix}0.4359 & -0.1490\\-0.1490 & 0.2223\end{pmatrix}$	$\begin{bmatrix}0.3528,\\0.5497\end{bmatrix}$ $\begin{bmatrix}-0.2310,\\-0.0830\end{bmatrix}$ $\begin{bmatrix}-0.2310,\\-0.0830\end{bmatrix}$ $\begin{bmatrix}0.1611,\\0.3089\end{bmatrix}$
σ_ϵ	0.0086111	$[0.008002, 0.009317]$	0.008613	$[0.00798, 0.009315]$

6.3.4.1.4. Links reliability estimation updating

The updated 10,000 MCMC draws from the joint posterior distributions of the model parameters were used in equation (115) to compute links reliability draws for 1,001 time values between 0 and 10,000 hours. The median values as well as the 95% credible bounds are shown in Figure 53. As expected, the informative prior Bayesian updating improved the precision of the estimates. Figure 54 shows both the initial links reliability estimation and credible intervals in blue solid lines. Figure 54 also shows the updated links reliability estimation along with credible intervals in red dashed lines. The updated reliability and credible intervals are shown from the time = 5,760 hours, when "new data" become available. The additional new data help to improve the credible interval with narrower band as compared to the initial credible interval. This is caused by reduction in uncertainty because of the availability of additional new data.

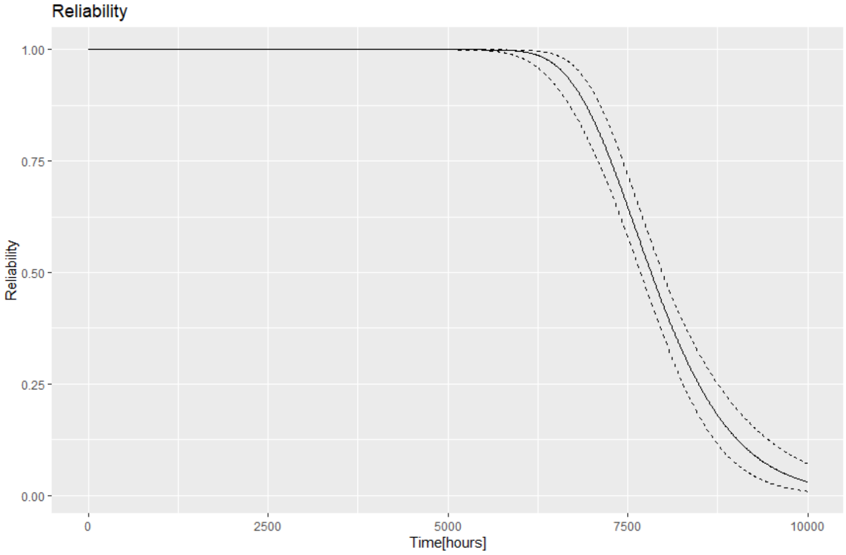


Figure 53. Updated links reliability and 95% credible intervals

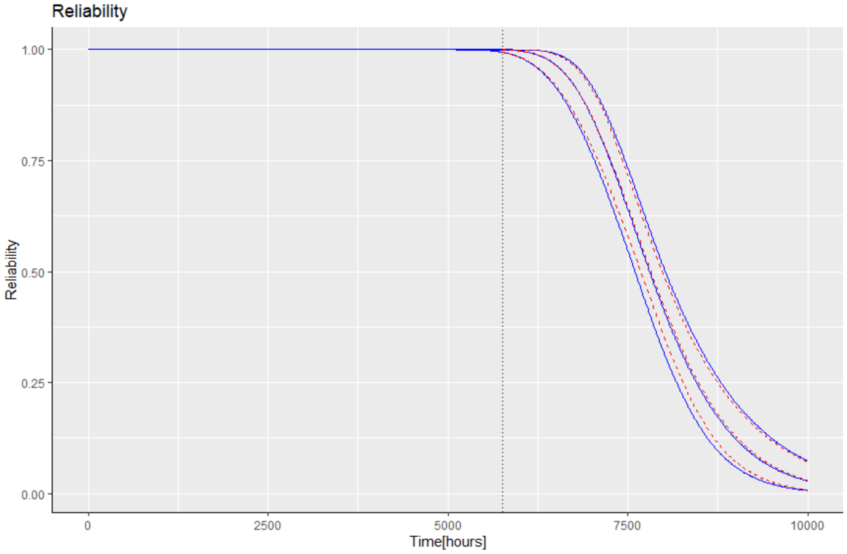


Figure 54. Initial (solid lines) and updated (dashed lines) links reliability and 95% credible intervals

6.3.4.2. Nodes degradation modeling for reliability evaluation

In this section, normalized (relative to an initial measurement taken on each unit) degradation data [282] for LED light-output are considered as shown in Figure 55. There is a decrease in light intensity output with time and LED failure is defined when the relative light intensity output reaches to 60% level of the initial value [282]. The sample degradation paths in the first 138 hours had a complicated irregular behavior for which LED experts had no explanation [282, 290]. Since, primary interest is in the long-run behavior of the LEDs, the first 138 hours of data were omitted. The remaining data were renormalized so that all the units start with a (normalized) output value of 1 at time = 138 hours. The truncated renormalized data are shown in Figure 56. The group at 130 °C junction temperature and 40 mA current is believed to cause the occurrence of a new failure mechanism [282], hence the degradation data of this group were removed before the parameters estimation. Moreover, from each of the remaining five groups, the degradation data of five sample units were removed so that the “initial data” for the analysis contain degradation paths for 125 sample units (five groups with 25 sample units per group) to match the 125 nodes of the case study network. Bayesian parameter estimation based on the “initial data” will be described in the next section (6.3.4.2.1). Additionally, using the initial estimated parameters, “new data” will be generated to demonstrate the Bayesian parameter updating in section 6.3.4.2.3.

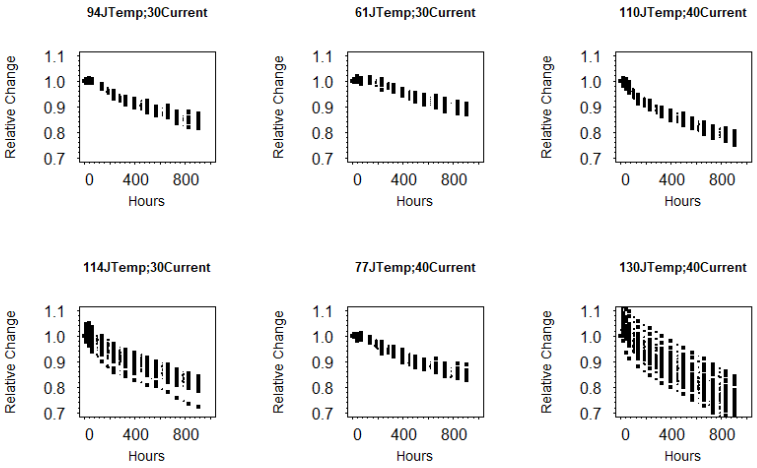


Figure 55. Original normalized LED degradation data

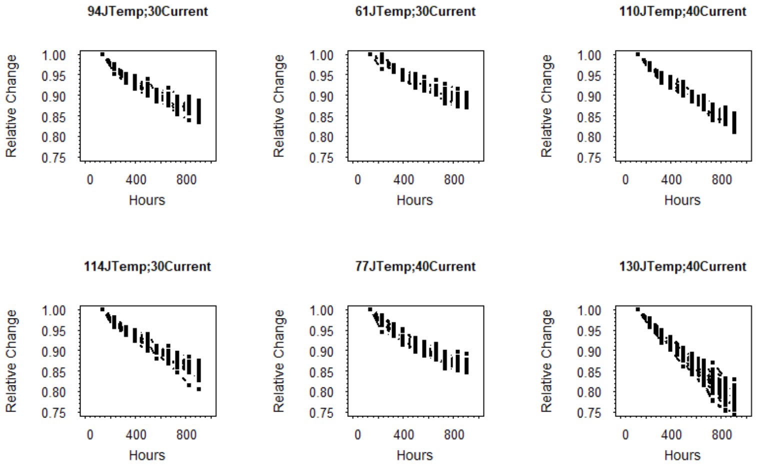


Figure 56. Renormalized LED degradation data

6.3.4.2.1. Bayesian estimation of parameters

The degradation model is given by equation (119) with random parameters b and ε. b can be described by a normal distribution with parameters (μ_b, σ_b), and the residual deviation ε is described by a normal distribution with mean zero and standard deviation parameter σ_ε. We propose Bayesian estimation of such parameters as well as of the parameters $\beta_{1,N}$ and $\beta_{2,N}$. In addition to degradation data, prior distributions are needed for the parameters $\mu_b, \sigma_b, \beta_{1,N}, \beta_{2,N}$, and σ_ε. Low-information distributions [290] will be considered for such parameters as shown in Table 24. “Flat” priors correspond to uniform distributions between $-\infty$ and ∞ [290].

Table 24. Low-information prior distribution specifications for nodes degradation model

Parameter	Prior distribution
μ_b	$Flat$
σ_b	$Uniform(1.0 \times 10^{-5}, 1.0 \times 10^{4})$
$\beta_{1,N}$	$Flat$
$\beta_{2,N}$	$Flat$
σ_ε	$Uniform(1.0 \times 10^{-5}, 1.0 \times 10^{4})$

A WinBUGS model was built considering the "initial data", the degradation model given by equation (119), the distributions assumed for the parameters b and ε, and the low-information prior distributions assumed parameters $\mu_b, \sigma_b, \beta_{1,N}, \beta_{2,N}$, and σ_ε given in Table 25. The initial 4,000 MCMC sample draws were dropped ("burn-in") and the sample draws were "thinned" to reduce autocorrelation, by setting a lag parameter L of 40. The point estimates obtained by the median values from the MCMC sample draws are: $\hat{\mu}_b = -0.008122$, $\widehat{\sigma}_b = 6.04 \times 10^{-4}$, $\beta_{1,N} = 3.961 \times 10^{-4}$, $\beta_{2,N} = -0.002555$, and $\hat{\sigma}_\epsilon = 0.004661$. These parameters will be used to simulate "new data" at normal operating conditions, defined by the test engineers as 40°C junction temperature and 20 mA current [282]. Figure 57 shows the kernel density estimations of parameter $\hat{\mu}_b$.

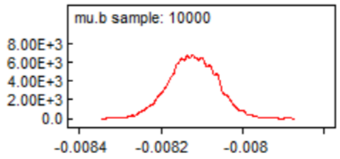


Figure 57. Bayesian kernel density estimation of parameter μ_b for "initial data"

6.3.4.2.2. Nodes reliability estimation

The 10,000 MCMC sample draws from the joint posterior distributions of the model parameters, were used in equation (126) to obtain nodes reliability draws for 1,001 time values between 0 and 10,000 hours. The median values (solid line) as well as the 95% credible bounds (dashed lines) are shown in Figure 58.

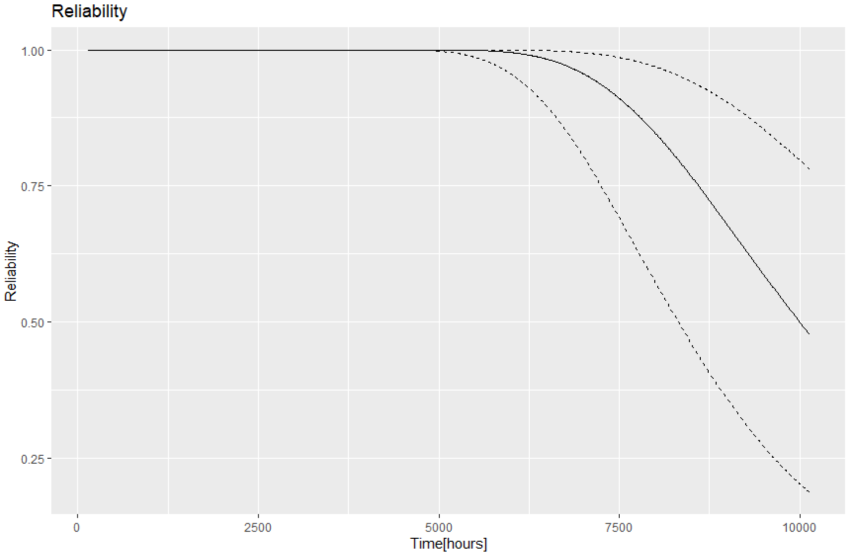


Figure 58. Nodes reliability and 95% credible intervals

6.3.4.2.3. Bayesian updating of parameters

"New data" were simulated considering time = 5,760 hours, 6,400 hours, 7,040 hours, and 7,680 hours, i.e., at the same time values as the "new data" for the links. A WinBUGS model was built considering the "new data", the degradation model given by equation (119), the distributions assumed for the parameters b and ε, and informative prior distributions assumed for parameters $\mu_b, \sigma_b, \beta_{1,N}$, and $\beta_{2,N}$. Initial MCMC sample draws obtained in section 6.3.4.2.1 were used to estimate the parameters of informative prior distributions (Table 25). The prior distributions for μ_b, $\beta_{1,N}$, and $\beta_{2,N}$ are specified with the MLE estimates obtained for normal distributions from the MCMC draws obtained in section 6.3.4.2.1 of μ_b , $\beta_{1,N}$, and $\beta_{2,N}$, respectively. On the other hand, the prior Gamma distribution for σ_b is defined by the MLE estimates is obtained from the MCMC sample draws obtained in section 6.3.4.2.1. Since this updating process is intended to take place with "new data" obtained during normal operation, which do not necessarily offer the same testing conditions as for “initial data”, a low-information prior is still considered for σ_{ε}.

The initial 4,000 MCMC sample draws were dropped (“burn-in”) and the sample draws were “thinned” to reduce autocorrelation, by setting a lag parameter L of 100. The point estimates obtained by the median values from the MCMC sample draws are: $\hat{\mu}_b = -0.008117$, $\widehat{\sigma}_b = 5.918 \times 10^{-4}$, $\beta_{1,N} = 4.019 \times 10^{-4}$, $\beta_{2,N} = -0.002658$, and $\hat{\sigma}_{\epsilon} = 0.004668$. Figure 59 shows the kernel density estimations of parameter $\hat{\mu}_b$.

Table 25. Informative prior distribution specifications for links degradation model

Parameter	Prior distribution
μ_b	$N(-0.008121648, 295558733^{-1})$
σ_b	$Gamma(220.2973{,}363870.2)$
$\beta_{1,N}$	$N(0.0003961045, 1012765573^{-1})$
$\beta_{2,N}$	$N(-0.002557142, 6333426^{-1})$
σ_{ε}	$Uniform(1.0 \times 10^{-5}, 1.0 \times 10^{4})$

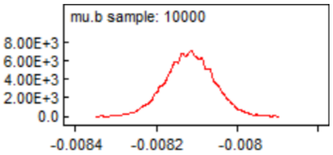


Figure 59. Bayesian kernel density estimation of parameter μ_b for “new data”

Like the nodes case, we performed sensitivity analysis to verify the robustness of the posterior distributions. Longer-tailed distributions was considered instead of normal distributions [292, 295]. Hence, in the alternative model, prior distributions for μ_b, $\beta_{1,N}$, and $\beta_{2,N}$ were specified with the MLE estimates obtained for Student’s t distributions from the MCMC draws obtained in section 6.3.4.2.1 of μ_b , $\beta_{1,N}$, and $\beta_{2,N}$, respectively. In WinBUGS, the (noncentral) Student’s t distribution is specified in terms of mean, precision (inverse of covariance), and degrees of freedom. In the alternative model, the prior distributions for μ_b, $\beta_{1,N}$, and $\beta_{2,N}$ are given as $t(-0.008121633, 301507501^{-1}, 100)$, $t(0.000396097, 1059545212^{-1}, 46)$, and $t(-0.002557149, 6460993^{-1}, 100)$, respectively. Table 26 shows the sensitivity of posterior inference in terms of the median and 95% credible intervals obtained from the MCMC sample draws. Minor differences are observed between the posteriors resulted by the model with normal distributions and the alternative model with t distributions. Therefore, the original model that considers the informative prior distributions in Table 25 was used for Bayesian updating of parameters.

Table 26. Posterior median and 95% credible intervals of parameters under different prior assumptions

Param.	Distributions for μ_b, $\beta_{1,N}$, and $\beta_{2,N}$			
	Normal		Student's t	
	Posterior median	95% posterior credible interval	Posterior median	95% posterior credible interval
μ_b	-0.008117	$[-0.008235, -0.008005]$	-0.008116	$[-0.008230, -0.008006]$
σ_b	5.918×10^{-4}	$[5.394 \times 10^{-4}, 6.504 \times 10^{-4}]$	5.910×10^{-4}	$[5.399 \times 10^{-4}, 6.480 \times 10^{-4}]$
$\beta_{1,N}$	4.019×10^{-4}	$[3.531 \times 10^{-4}, 4.516 \times 10^{-4}]$	4.023×10^{-4}	$[3.543 \times 10^{-4}, 4.529 \times 10^{-4}]$
$\beta_{2,N}$	-0.002658	$[-0.003160, -0.002164]$	-0.002648	$[-0.003141, -0.002137]$
σ_ε	0.004668	$[0.004353, 0.005016]$	0.004670	$[0.004350, 0.005034]$

6.3.4.2.4. Nodes reliability estimation updating

The updated 10,000 MCMC draws from the joint posterior distributions of the model parameters were used in equation (126) to compute links reliability draws for 1,001 time values between 0 and 10,000 hours. The median values (solid line) as well as the 95% credible bounds (dashed lines) are shown in Figure 60. As expected, the informative prior Bayesian updating improved the precision of the estimates. Figure 61 shows the initial nodes reliability estimation with credible intervals in blue solid lines. Figure 61 also shows the updated nodes reliability estimation along with credible intervals in red dashed lines. The updated reliability and credible intervals start at time = 5,760 hours, i.e., when “new data” become available. Credible intervals for updated nodes reliability are narrower than the initial credible intervals.

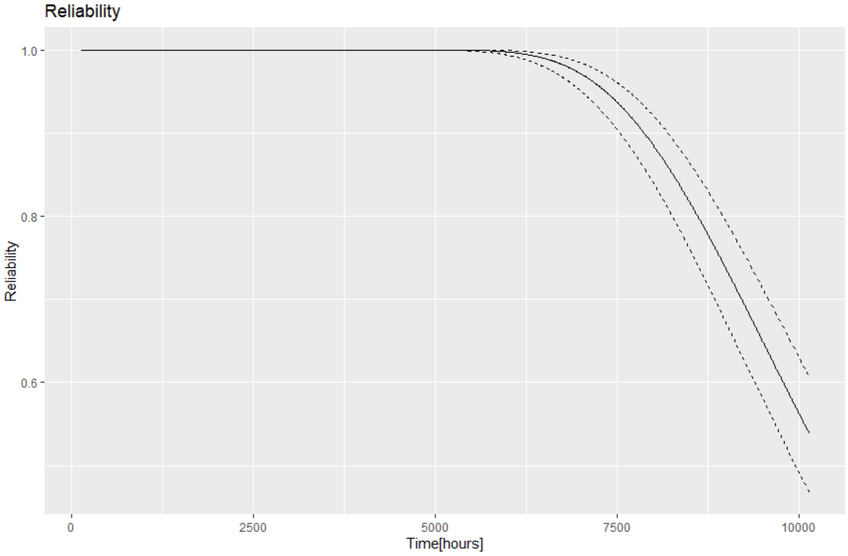


Figure 60. Updated nodes reliability and 95% credible intervals

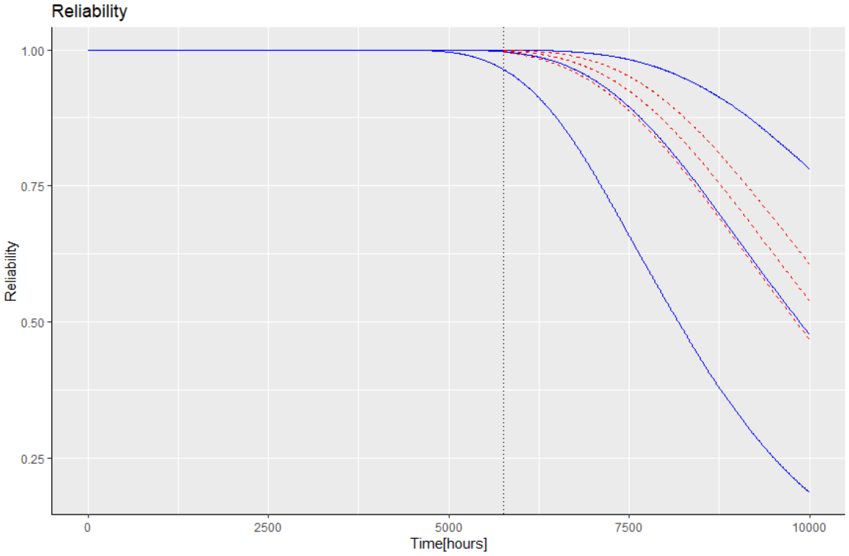


Figure 61. Initial and updated nodes reliability and 95% credible intervals

6.3.4.3. Monte Carlo method and deep neural network model for all-terminal network reliability estimation

In this section, the use of MC method and DNN to evaluate the network reliability is illustrated. MC method (Algorithm 6.1) is applied to the selected Ion network to obtain a set of estimated reliability values of links ($R_{MC_{links}}$). $R_{MC_{links}}$ will be the target during the training process, i.e., the y_t values. The best trained DNN is expected to estimate the network reliability of links ($\hat{R}_{links}$) for any given value of p_L. Finally, the all-terminal network reliability will be calculated by using equation (106) for any given p_L and p_N.

A dataset of 100 link reliability values is considered, i.e., $\{0.01, 0.02, \dots, 0.99, 1.00\}$. Based on this set of link reliability values, a base dataset of pairs (X_t, y_t) is formed. X_t is the links reliability value (p_L) and y_t is corresponding estimated reliability of links ($R_{MC_{links}}$), for each element in the set of link reliability values. The base dataset is divided in training and testing datasets by applying five-fold cross-validation.

The DNN architecture has two hidden layers. Different number of neurons ({5, 10, 20, 30, 40, 50}) were investigated for each hidden layer [267]. The dropout probability values from the set {0, 0.05, 0.10, 0.15, 0.20, 0.25}, where 0 indicates no dropout were employed. This provides a total of 216 experiments (six numbers of neurons in the first layer, six numbers of neurons in the second layer, and six dropout values). The average root mean square error (RMSE) considering cross-validation [253] is used to select the best DNN architecture. The best architecture is (5, 30, 0.15), i.e., 5 and 30 neurons in the first and second hidden layers, respectively, with a dropout of 0.15. The final application DNN is trained using all the 100 members of the data set and its validation error is inferred using the average cross-validation error (Table 27, column 2). The average cross validation-error is given by equation (107) [42, 75].

$$RMSE_{cv=}\sqrt{\frac{1}{100}\sum_{g=1}^{5}\sum_{h=1}^{20}\left(y_{(g-1)\times 20+h}-\hat{y}_{(g-1)\times 20+h}\right)^{2}} \tag{136}$$

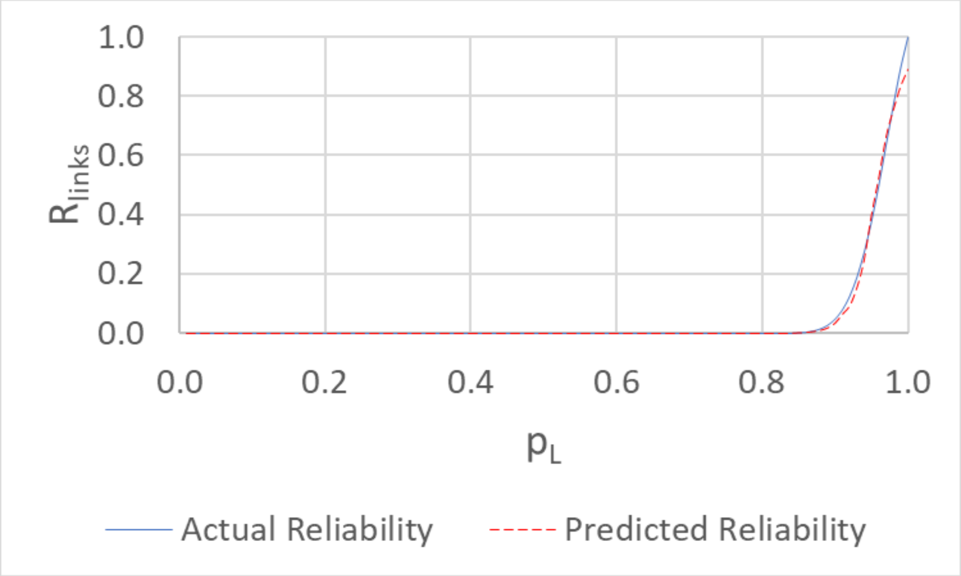


Figure 62. Estimated reliability of links ($\hat{R}_{links}$), as a function of links reliability (p_L)

Table 27. MC and DNN performance

Architecture	Error	Paired t test		Computation time	
Best DNN	RMSE	p-value	95% C.I.	Monte Carlo [s]	MC-DNN model [ms]
5, 30, 0.15	0.01460	0.1029	[-0.0005, 0.0053]	1223.59	0.316

The error (0.01460) outperforms previous results achieved by ANN-based approaches, e.g., RMSE of 0.06260 [42] and RMSE of 0.04406 [75]. Also, a paired t-test between the actual reliability and the reliability predicted by the DNN was performed. p-values and 95% confidence intervals (Table 27, columns 3, 4) for the mean difference show no significant pairwise difference between actual and the predicted values. Therefore, the DNN provides good fit as shown in Figure 62. Figure 62 also shows that the predicted values (R_{links}) noticeably underestimate the actual reliability when the links reliability values (p_L) are greater than 0.99 (approximately). To improve this performance, a hierarchical approach that integrates a specialized DNN trained for link reliabilities greater than 0.99, is used. The best specialized DNN architecture was (50, 50, 0). Therefore, the appropriate DNN should be selected in equation (106) when applied for network reliability estimation.

There is a significant computation time difference between the pure MC algorithm and the integrated framework based on MC-DNN (Table 27, columns 5, 6). Table 27 (columns 5, 6) shows the computation time of a single network reliability value $\hat{R}_{net}$, for a given combination of input pair of values (p_L, p_N). The MC average time calculation (1223.59 s) for a single value $\hat{R}_{net}$ is based on the total time to estimate 100 link reliability values. On the other hand, the total time to compute the network reliability for 10,000 input pairs (links and nodes reliability draws) for a total of 1,001 time values (between 0 and 10,000 hours) was 3,162 seconds. However, the DNN model average computation time for a single value of $\hat{R}_{net}$ is only 0.316 ms (i.e., $\frac{3{,}162s}{10{,}000 \times 1{,}001}$). This time reduction is convenient for fast reliability estimation as in approximately 3 seconds, 10,000 network reliability draws can be obtained providing not only a point estimate but also credible bounds for any given time value. A laptop with a processor Intel(R) Core (TM) i7-8565U CPU @ 1.80GHz, and 16GB in RAM was used.

6.3.4.4. Network reliability estimation

In this section the network reliability estimation is illustrated. The links reliability draws (section 6.3.4.1.2) and the nodes reliability draws (section 6.3.4.2.2) were fed to the hierarchical DNN model (equation (106) with the appropriate DNN) to obtain network reliability draws for

1,001 time values between 0 and 10,000 hours. The median values (solid line) as well as the 95% credible bounds (dashed lines) are shown in Figure 63.

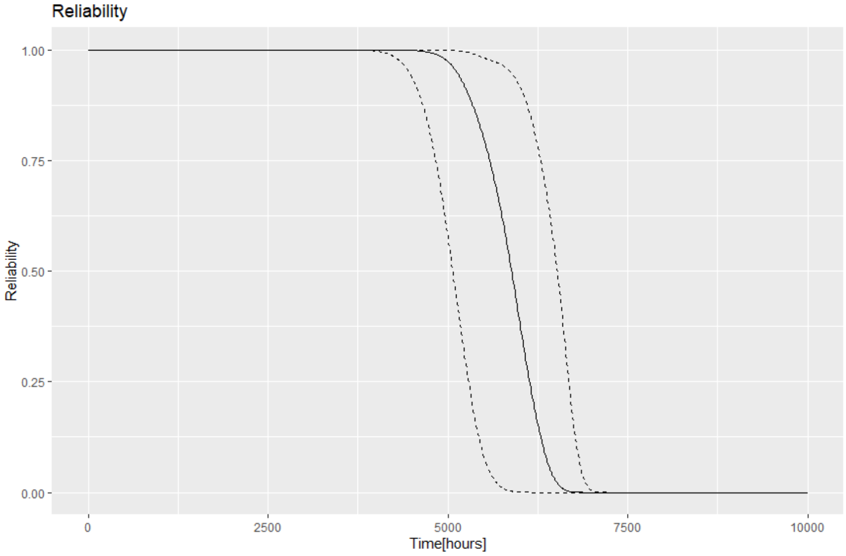


Figure 63. Network reliability and 95% credible intervals

6.3.4.5. Network reliability estimation updating

Once additional information becomes available, the updated links reliability draws (discussed in section 6.3.4.1.4) and nodes reliability draws (discussed in section 6.3.4.2.4) are fed to the proposed DNN model to obtain updated network reliability draws for the corresponding 1,001 time values between 0 and 10,000 hours. The median values (solid line) as well as the 95% credible bounds (dashed lines) are shown in Figure 64. As expected, the informative prior Bayesian updating improved the precision of the estimates. Figure 65 shows the initial network reliability estimation with credible intervals in blue solid lines. Figure 65 also shows the updated network reliability estimation with credible intervals in red dashed lines. The updated reliability and credible intervals start at time = 5,760 hours, i.e., when “new data” become available. Credible intervals for updated network reliability are narrower than the initial credible intervals.

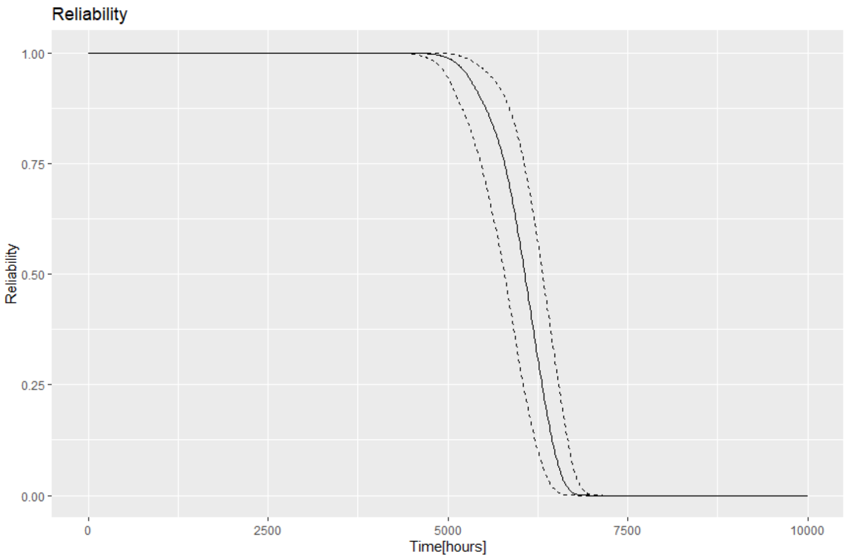


Figure 64. Updated network reliability and 95% credible intervals

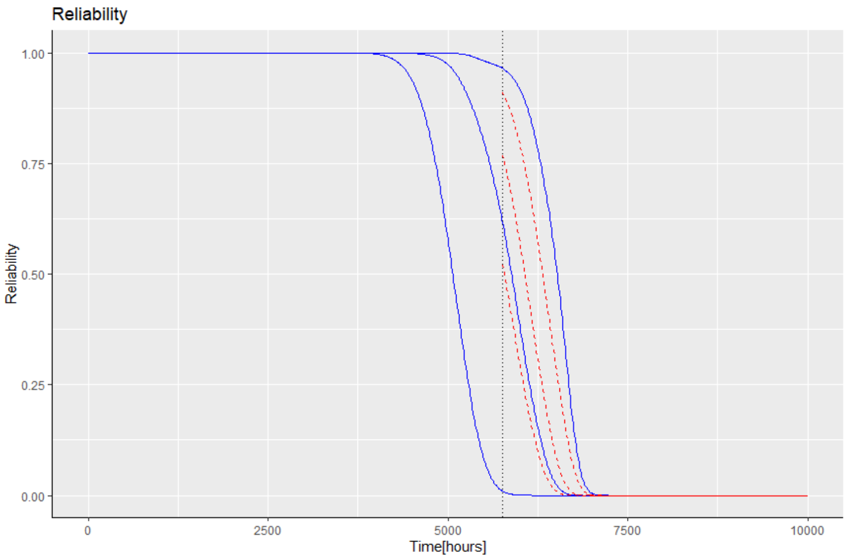


Figure 65. Initial and updated network reliability and 95% credible intervals

6.3.5. Conclusion

Most of the current work on network reliability considers perfect nodes and that the reliability of links or nodes is constant or even perfect. This study has considered the reliability of both links and nodes as a function of time in the prediction of all-terminal network reliability. This work has proposed a framework to account the dynamic behavior of network by using degradation data from both links and nodes of a network and to estimate its all-terminal reliability as a function of time. Due to the complexity of the problem, the proposed framework integrates BM, MC simulation, and DNN. BM allows both initial estimation of degradation model parameters and updating of parameters with new data. Links and nodes reliability estimates can be evaluated from the model parameters. In addition, an integration of MC-DNN with Bayesian approach provides accurate and fast estimation of both initial and updated predictions of links/nodes and/or all-terminal network reliability functions for any given time, not only as point estimates but as credible intervals.

The proposed framework could be used in situations where fast links, nodes, and/or network reliability estimation and updating is required, such as an online reliability monitoring system. Based on (usually limited) initial accelerated degradation test data, the framework could provide estimates of reliability as a function of time. Furthermore, if during normal operation, links, nodes, or both change their degradation profile, this variation in new data can be captured by the framework for proper and timely updating of the reliability predictions. Therefore, the proposed framework is compatible with and provides a way to take advantage of modern sensors technology as sources of new degradation data to update the reliability predictions. The updated reliability predictions may provide valuable information to decision makers for taking proper actions regarding network operations management. This information is important for the users as well as for the manufacturers, especially in logistical decision making such as preventive maintenance, warranty policy, and spare parts management.

CHAPTER 7. CONCLUSION AND SUMMARY

This dissertation presents a series of studies on the reliability analysis of complex CPSs. Reliability modeling and assessment approaches are provided considering physical components, hardware/software interactions, and overall CPS reliability modeled as networks.

A generic model is proposed for reliability prediction of physical components of CPS. The proposed model considers multi-stress variables and multi-stress levels and is based on ADT test design and data analysis. An extensive review on FHE testing methods provides input for the ADT test design FHE. Although ADT data of FHE devices were used to demonstrate the developed ADT methods, most of them, with careful consideration of material properties, could be applied to other kind of physical components that exhibit degradation. The proposed model considers multi-stress factors combinations experiments whose resultant degradation data are used to fit a nonlinear mixed-effects regression model and estimate time-to-failure distributions parameters. The mean-time-to-failure estimates for different stress combinations are then used to fit a developed multi-stress factor stress-life (S-L) model. The S-L model, based on equivalent stress, allows estimating the product-life under given operating conditions. The framework also provides a method to estimate both the expected damage accumulation and its variability at any given time and operating conditions. Additionally, the framework provides a procedure to evaluate the reliability as a function of time and given operating conditions, useful for both constant and sequences of (varying) operating conditions.

Since hardware, software, and their interactions constitute crucial aspects of CPSs, a reliability modeling and assessment approach is proposed to take into account such aspects. Most of the existing work has assumed either independence between hardware (HW) and software (SW) or a fixed proportion of hardware reported failures to represent HW/SW interactions. On the other hand, this work proposes a reliability model for a system that captures the changing interactions between hardware and software based on probabilistic models. The randomness of the fraction of HW/SW interaction failures demands the use of stochastic programming methods to estimate the parameters of a hardware failure time distribution and the corresponding hardware reliability. To consider randomness, a stochastic optimization problem is formulated and solved with the general algebraic modeling system. Software reliability is modeled with a non-homogeneous Poisson process (NHPP) to fit the software failure data. To capture the interaction between hardware and software, a Markov process model is used by considering that a degraded hardware state leads to a HW/SW failure and eventually to a system failure. By considering this probabilistic approach, confidence intervals and quartiles can be obtained for system reliability in addition to point estimators.

Complex CPSs can be modeled by networks, whose reliability estimation is a NP-hard problem. As an alternative to classical exact NP-hard algorithms, approaches based on deep learning are receiving attention. This dissertation proposes a CNN-based approach to estimate the all-terminal network reliability. This approach introduces a multidimensional matrix format to embed the topological and link reliability information of networks, providing a novel use of CNNs beyond image classification. Moreover, a DNN-based approach is proposed by integrating DNN and GEM. This integration allows the estimation of all-terminal network reliability considering varying network sizes. Neither proposed approach needs the reliability upper-bound as an input and outperform previous works in terms of the RMSE. Nevertheless, such approaches consider perfect nodes.

In addition, this dissertation proposes a DNN-MC approach to estimate the reliability of a network considering that both links and nodes can fail. The proposed DNN-MC framework can be used for real-time monitoring of network reliability.

Finally, the relaxation of perfect nodes assumption as well as the consideration of degradation in the components of a network, makes a contribution to realistic reliability estimation of networks. Hence, this dissertation develops a framework to estimate the all-terminal reliability of a network, whose nodes and links not only have the possibility to fail but also exhibit degradation. Degradation data provides useful information to estimate the all-terminal network reliability as a function of time. Due to the complexity of the problem, the proposed framework integrates BM, MC, and DNN. BM is employed for both initial estimation of degradation model parameters and updating of parameters as new data become available. The case study results revealed that updating of parameter allows to reduce the uncertainty in parameters estimation by incorporating new data to the proposed framework. Moreover, the framework provides credible intervals on top of point estimates of reliability for links, nodes, and network. The initial and updated reliability predictions may provide valuable information to decision makers for taking proper actions regarding network operations management. This information is important for the users as well as for the manufacturers, especially in logistical decision making such as preventive maintenance, warranty policy, and spare parts management.

7.1. Future research

The proposed framework provides a way to take advantage of modern sensors technology that can capture multidimensional data in real-time. Such data is useful for network reliability estimation and updating of parameters.

After the proposed dissertation, future research direction may include areas such as the following:

- MLE estimation of parameter was applied at component level in CHAPTER 3. Bayesian methods could be explored as well for parameter estimation and updating if new data are available. Bayesian methods would allow to obtain credible intervals on top of point estimates.
- General degradation path model has been considered in this work at both component and system level. Stochastic processes, e.g., Wiener or gamma processes, could be investigated as well.
- This work has assumed distributions for parameters for the hardware/software interactions model, based on the literature. To improve the accuracy, root cause analysis databases could provide statistical data on the proportion of failures classified as hardware that are actually HW/SW interactions failures. Such data could be used to provide more realistic estimate of the assumed distributions.
- Deep learning methods such as DNN and CNN have been considered for all-terminal network reliability estimation. More sophisticated approaches could be investigated, e.g., graph neural networks. An abstract about this research idea was recently submitted for RAMS 2022 Conference.
- On top of the estimation of network reliability as a function of time proposed in CHAPTER 6, the RAUL can be estimated as well. RUL estimation could be based on the network reliability and a given threshold reliability for the network. RUL prediction can be a useful input for prognostic health management.

REFERENCES

[1] E. A. Lee and S. A. Seshia, *Introduction to Embedded Systems: A Cyber-Physical Systems Approach*. MIT Press, 2017.

[2] National Science Foundation. "Cyber-Physical Systems (CPS) Program Solicitation NSF 19-553." National Science Foundation. https://www.nsf.gov/publications/pub_summ.jsp?WT.z_pims_id=503286&ods_key=nsf19553 (accessed 7/29/2019, 2019).

[3] K. Marashi, S. S. Sarvestani, and A. R. Hurson, "Consideration of cyber-physical interdependencies in reliability modeling of smart grids," *IEEE Transactions on Sustainable Computing,* vol. 3, no. 2, pp. 73-83, 2017.

[4] Z. Li and R. Kang, "Strategy for reliability testing and evaluation of cyber physical systems," in *2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2015: IEEE, pp. 1001-1006.

[5] S. Rebello, H. Yu, and L. Ma, "An integrated approach for system functional reliability assessment using Dynamic Bayesian Network and Hidden Markov Model," *Reliability Engineering & System Safety,* vol. 180, pp. 124-135, 2018.

[6] B. Liscouski and W. Elliot, "Final report on the august 14, 2003 blackout in the united states and canada: Causes and recommendations," *A report to US Department of Energy,* vol. 40, no. 4, p. 86, 2004.

[7] J. Glotfelty, "Transforming the grid to revolutionize electric power in north america," *Tech. Rep.,* 2003.

[8] S. M. Rinaldi, J. P. Peerenboom, and T. K. Kelly, "Identifying, understanding, and analyzing critical infrastructure interdependencies," *IEEE control systems magazine,* vol. 21, no. 6, pp. 11-25, 2001.

[9] M. J. Assante, "Confirmation of a coordinated attack on the Ukrainian power grid," *SANS Industrial Control Systems Security Blog,* vol. 207, 2016.

[10] D. Wei, Y. Lu, M. Jafari, P. M. Skare, and K. Rohde, "Protecting Smart Grid Automation Systems Against Cyberattacks," *IEEE Transactions on Smart Grid,* vol. 2, no. 4, pp. 782-795, 2011.

[11] M. Korkali, J. G. Veneman, B. F. Tivnan, J. P. Bagrow, and P. D. H. Hines, "Reducing Cascading Failure Risk by Increasing Infrastructure Network Interdependence," vol. 7, p. 44499, 2017.

[12] T. Staff. "Steinitz Israel's electric authority hit by 'severe' cyber-attack." https://www.timesofisrael.com/steinitz-israels-electricauthority-hit-by-severe-cyber-attack/ (accessed August 5, 2019).

[13] L. Shi, Q. Dai, and Y. Ni, "Cyber–physical interactions in power systems: A review of models, methods, and applications," vol. 163, pp. 396-412, 2018.

[14] W. Q. Meeker and L. A. Escobar, *Statistical Methods for Reliability Data*. New york: John Wiley & Sons, Inc., 1998.

[15] NextFlex. "Defining Flexible Hybrid Electronics." https://www.nextflex.us/about/about-fhe/ (accessed 5/2, 2020).

[16] N. Wu and X. Li, "RFID applications in cyber-physical system," *RFID Applications in Cyber-Physical System,* pp. 291-302, 2011.

[17] A. Akanmu, C. Anumba, and J. Messner, "Scenarios for cyber-physical systems integration in construction," *Journal of Information Technology in Construction (ITcon),* vol. 18, no. 12, pp. 240-260, 2013.

[18] E. A. Lee, "Computing foundations and practice for cyber-physical systems: A preliminary report," *University of California, Berkeley, Tech. Rep. UCB/EECS-2007-72,* vol. 21, 2007.

[19] E. A. Lee, "Cyber physical systems: Design challenges," in *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, 2008: IEEE, pp. 363-369.

[20] E. Lee, "CPS Foundations in Proceedings of Design Automation Conference (DAC)," ed: ACM, 2010.

[21] A. Huebner, C. Facchi, M. Meyer, and H. Janicke, "RFID systems from a cyber-physical systems perspective," in *2013 Proceedings of the 11th Workshop on Intelligent Solutions in Embedded Systems (WISES)*, 2013: IEEE, pp. 1-6.

[22] X. Teng, H. Pham, and D. R. Jeske, "Reliability Modeling of Hardware and Software Interactions, and Its Applications," *IEEE Transactions on Reliability,* vol. 55, no. 4, pp. 571 - 577, 2006.

[23] D. S. Roy, S. Verma, C. Murthy, and D. K. Mohanta, "Reliability Assessment of Cyber-Physical Systems: A Hardware–Software Interaction Perspective," in *Cyber-Physical Systems: A Computational Perspective*. Boca Raton: Taylor & Francis Group, LLC, 2016, pp. 249-260.

[24] E. Feng, J. Zheng, and C. Liu, "An integrated reliability model of hardware-software system," in *2014 10th International Conference on Reliability, Maintainability and Safety (ICRMS),* 6-8 Aug. 2014 2014, pp. 577-580.

[25] H. Koc, S. S. Shaik, and P. P. Madupu, "Reliability Modeling and Analysis for Cyber Physical Systems," in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC),* 2019: IEEE, pp. 0448-0451.

[26] J. Park, H.-J. Kim, J.-H. Shin, and J. Baik, "An embedded software reliability model with consideration of hardware related software failures," in *2012 IEEE Sixth International Conference on Software Security and Reliability,* 2012: IEEE, pp. 207-214.

[27] B. Schroeder and G. Gibson, "A large-scale study of failures in high-performance computing systems," *IEEE transactions on Dependable and Secure Computing,* vol. 7, no. 4, pp. 337-350, 2009.

[28] B. Huang, X. Li, M. Li, J. Bernstein, and C. Smidts, "Study of the impact of hardware fault on software reliability," in *16th IEEE International Symposium on Software Reliability Engineering (ISSRE'05)*, 2005: IEEE, pp. 10 pp.-72.

[29] R. K. Wood, "Factoring algorithms for computing K-terminal network reliability," *IEEE Transactions on Reliability,* vol. 35, no. 3, pp. 269-278, 1986.

[30] J. Ayoub, W. Saafin, and B. Kahhaleh, "K-terminal reliability of communication networks," in *ICECS 2000. 7th IEEE International Conference on Electronics, Circuits and Systems (Cat. No. 00EX445),* 2000, vol. 1: IEEE, pp. 374-377.

[31] H. Cancela, M. El Khadiri, and L. A. Petingi, "Polynomial-time topological reductions that preserve the diameter constrained reliability of a communication network," *IEEE Transactions on Reliability,* vol. 60, no. 4, pp. 845-851, 2011.

[32] Z. Zhang, W. An, and F. Shao, "Cascading failures on reliability in cyber-physical system," *IEEE Transactions on Reliability,* vol. 65, no. 4, pp. 1745-1754, 2016.

[33] J. Johansson and H. Hassel, "An approach for modelling interdependent infrastructures in the context of vulnerability analysis," *19th European Safety and Reliability Conference,* vol. 95, no. 12, pp. 1335-1344, 2010.

[34] W. Liu, Q. Gong, H. Han, Z. Wang, and L. Wang, "Reliability modeling and evaluation of active cyber physical distribution system," *IEEE Transactions on Power Systems,* vol. 33, no. 6, pp. 7096-7108, 2018.

[35] H. Gunduz and D. Jayaweera, "Reliability assessment of a power system with cyber-physical interactive operation of photovoltaic systems," vol. 101, pp. 371-384, 2018.

[36] Y. n. Wang, Z. y. Lin, X. Liang, W. y. Xu, Q. Yang, and G. f. Yan, "On modeling of electrical cyber-physical systems considering cyber security," *Frontiers of Information Technology & Electronic Engineering,* journal article vol. 17, no. 5, pp. 465-478, May 01 2016, doi: 10.1631/fitee.1500446.

[37] M. Parandehgheibi, E. Modiano, and D. Hay, "Mitigating cascading failures in interdependent power grids and communication networks," in *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2014: IEEE, pp. 242-247.

[38] J. Kim and L. Tong, "On topology attack of a smart grid: Undetectable attacks and countermeasures," *IEEE Journal on Selected Areas in Communications,* vol. 31, no. 7, pp. 1294-1305, 2013.

[39] A. M. Shooman and A. Kershenbaum, "Exact graph-reduction algorithms for network reliability analysis," in *IEEE Global Telecommunications Conference GLOBECOM'91: Countdown to the New Millennium. Conference Record*, 1991: IEEE, pp. 1412-1420.

[40] M. L. Shooman, *Reliability of computer systems and networks: fault tolerance, analysis, and design*. John Wiley & Sons, 2003.

[41] J.-M. Won and F. Karray, "Cumulative update of all-terminal reliability for faster feasibility decision," *IEEE Transactions on reliability,* vol. 59, no. 3, pp. 551-562, 2010.

[42] C. Srivaree-Ratana, A. Konak, and A. E. Smith, "Estimation of all-terminal network reliability using an artificial neural network," *Computers & Operations Research,* vol. 29, no. 7, pp. 849-868, 2002.

[43] D. R. Karger, "A randomized fully polynomial time approximation scheme for the all-terminal network reliability problem," *SIAM review,* vol. 43, no. 3, pp. 499-522, 2001.

[44] J. E. Ramirez-Marquez and C. M. Rocco, "All-terminal network reliability optimization via probabilistic solution discovery," *Reliability Engineering & System Safety,* vol. 93, no. 11, pp. 1689-1697, 2008.

[45] F. Altiparmak, B. Dengiz, and A. E. Smith, "A general neural network model for estimating telecommunications network reliability," *IEEE transactions on reliability,* vol. 58, no. 1, pp. 2-9, 2009.

[46] A. Konak and A. E. Smith, "An improved general upperbound for all-terminal network reliability," in *Industrial Eng. Res. Conf*, 1998.

[47] M.-S. Yeh, "A new Monte Carlo method for estimating network reliability," in *Proceedings of the 16th International Conference on Computers & Industrial Engineering, 1994*, 1994.

[48] J. E. Ramirez-Marquez and D. W. Coit, "A Monte-Carlo simulation approach for approximating multi-state two-terminal reliability," *Reliability Engineering & System Safety,* vol. 87, no. 2, pp. 253-264, 2005.

[49] G. S. Fishman, "A Monte Carlo sampling plan for estimating network reliability," *Operations Research,* vol. 34, no. 4, pp. 581-594, 1986.

[50] J. B. Kruskal, "The number of simplices in a complex," *Mathematical optimization techniques,* vol. 10, pp. 251-278, 1963.

[51] G. Katona, "A theorem of finite sets, Theory of graphs (Proc. Colloq., Tihany, 1966)," ed: Academic Press, New York, 1968.

[52] S. M. Ross, *Introduction to probability models*. Academic press, 2014.

[53] S. Nannapaneni, S. Mahadevan, S. Pradhan, and A. Dubey, "Towards reliability-based decision making in cyber-physical systems," in *2016 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2016: IEEE, pp. 1-6.

[54] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin, "Catastrophic cascade of failures in interdependent networks," vol. 464, p. 1025, 2010.

[55] F. Moskowitz, "The analysis of redundancy networks," *Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics,* vol. 77, no. 5, pp. 627-632, 1958.

[56] Z. Wu, N. Huang, X. Zheng, and X. Li, "Cyber-physical avionics systems and its reliability evaluation," in *The 4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent*, 2014: IEEE, pp. 429-433.

[57] K. Marashi and S. S. Sarvestani, "Towards comprehensive modeling of reliability for smart grids: Requirements and challenges," in *2014 IEEE 15th International Symposium on High-Assurance Systems Engineering*, 2014: IEEE, pp. 105-112.

[58] X. Sun, N. Huang, B. Wang, and J. Zhou, "Reliability of cyber physical systems assessment of the aircraft fuel management system," in *The 4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent*, 2014: IEEE, pp. 424-428.

[59] C. J. Lu and W. Q. Meeker, "Using Degradation Measures to Estimate a Time-to-Failure Distribution," *Technometrics,* vol. 35, no. 2, pp. 161-174, 1993.

[60] S. Tang, X. Guo, C. Yu, H. Xue, and Z. Zhou, "Accelerated degradation tests modeling based on the nonlinear Wiener process with random effects," *Mathematical Problems in Engineering,* vol. 2014, 2014.

[61] J. Tang and T. S. Su, "Estimating failure time distribution and its parameters based on intermediate data from a Wiener degradation model," *Naval Research Logistics (NRL),* vol. 55, no. 3, pp. 265-276, 2008.

[62] Z.-S. Ye, Y. Wang, K.-L. Tsui, and M. Pecht, "Degradation data analysis using Wiener processes with measurement errors," *IEEE Transactions on Reliability,* vol. 62, no. 4, pp. 772-780, 2013.

[63] Z. Pan and N. Balakrishnan, "Reliability modeling of degradation of products with multiple performance characteristics based on gamma processes," *Reliability Engineering & System Safety,* vol. 96, no. 8, pp. 949-957, 2011.

[64] J. Lawless and M. Crowder, "Covariates and random effects in a gamma process model with application to degradation and failure," *Lifetime Data Analysis,* vol. 10, no. 3, pp. 213-227, 2004.

[65] S. Limon, O. P. Yadav, and B. Nepal, "Estimation of Product Lifetime Considering Gamma Degradation Process with Multi-Stress Accelerated Test Data," in *IIE Annual Conference. Proceedings,* 2017: Institute of Industrial and Systems Engineers (IISE), pp. 1387-1392.

[66] L. Wang, R. Pan, X. Li, and T. Jiang, "A Bayesian reliability evaluation method with integrated accelerated degradation testing and field information," *Reliability Engineering & System Safety,* vol. 112, pp. 38-47, 2013.

[67] S. Limon, O. P. Yadav, and B. Nepal, *Remaining Useful Life Prediction Using ADT data with Inverse Gaussian Process Model*. 2018.

[68] W. Peng, Y.-F. Li, Y.-J. Yang, H.-Z. Huang, and M. J. Zuo, "Inverse Gaussian process models for degradation analysis: A Bayesian perspective," *Reliability Engineering & System Safety,* vol. 130, pp. 175-189, 2014.

[69] F. Altiparmak, B. Dengiz, and A. E. Smith, "Reliability optimization of computer communication networks using genetic algorithms," in *SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 98CH36218),* 1998, vol. 5: IEEE, pp. 4676-4681.

[70] A. Davila-Frias, S. Limon, V. Marinov, and O. P. Yadav, "Reliability Evaluation of Flexible Hybrid Electronics Systems Considering Degradation Behavior Under Multistress Operating Conditions," *Journal of Electronic Packaging,* vol. 143, no. 2, 2020, doi: 10.1115/1.4048035.

[71] A. Davila-Frias, O. P. Yadav, and V. Marinov, "A Review of Methods for the Reliability Testing of Flexible Hybrid Electronics," *IEEE Transactions on Components, Packaging and Manufacturing Technology,* 2020.

[72] A. Davila-Frias, V. Marinov, O. P. Yadav, and Y. Atanasov, "Design of Accelerated Degradation Test Method and Failure Analysis of Flexible Hybrid Electronic Devices," in *International Electronic Packaging Technical Conference and Exhibition*, 2020, vol. 84041: American Society of Mechanical Engineers, p. V001T02A001.

[73] A. Davila Frias, N. Yodo, and O. P. Yadav, "Mixed-Degradation Profiles Assessment of Critical Components in Cyber-Physical Systems," in *2019 Annual Reliability and Maintainability Symposium (RAMS),* 28-31 Jan. 2019 2019, pp. 1-6.

[74] A. Davila-Frias, N. Yodo, and O. Yadav, "Probabilistic modeling of hardware and software interactions for reliability prediction of embedded systems," 2019.

[75] A. Davila-Frias and O. P. Yadav, "All-terminal network reliability estimation using convolutional neural networks," *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability,* p. 1748006X20969465, 2020.

[76] A. Davila-Frias, S. Salem, and O. P. Yadav, "Deep Neural Networks (DNNs) For All-Terminal Network Reliability Estimation," 2020.

[77] A. Davila-Frias, O. P. Yadav, S. Salem, and B. Nepal, "All-Terminal Network Reliability Estimation with Monte Carlo and Deep Neural Networks," 2021.

[78] A. Davila-Frias, N. Yodo, T. Le, and O. P. Yadav, "A Deep Neural Network and Bayesian Method based Framework for All-Terminal Network Reliability Estimation Considering Degradation," 2021.

[79] E. Zio, *An introduction to the basics of reliability and risk analysis*. World scientific, 2007.

[80] D. L. Banks, "Foundations of Risk Analysis: A Knowledge and Decision-Oriented Perspective. Terje Aven," *Journal of the American Statistical Association,* vol. 100, pp. 703-704, 2005.

[81] M. Rausand and A. Høyland, *System reliability theory: models, statistical methods, and applications*. John Wiley & Sons, 2003.

[82] J. Guo, Y. Han, C. Guo, F. Lou, and Y. Wang, "Modeling and vulnerability analysis of cyber-physical power systems considering network topology and power flow properties," *Energies,* vol. 10, no. 1, p. 87, 2017.

[83] X. Ji *et al.*, "Will electrical cyber–physical interdependent networks undergo first-order transition under random attacks?," *Physica A: Statistical Mechanics and its Applications,* vol. 460, pp. 235-245, 2016.

[84] Z. Huang, C. Wang, S. Ruj, M. Stojmenovic, and A. Nayak, "Modeling cascading failures in smart power grid using interdependent complex networks and percolation theory," in *2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)*, 2013: IEEE, pp. 1023-1028.

[85] J. Hu, J. Yu, J. Cao, M. Ni, and W. Yu, "Topological interactive analysis of power system and its communication module: A complex network approach," *Physica A: Statistical Mechanics and its Applications,* vol. 416, pp. 99-111, 2014.

[86] J. Yan, H. He, and Y. Sun, "Integrated security analysis on cascading failure in complex networks," *IEEE Transactions on Information Forensics and Security,* vol. 9, no. 3, pp. 451-463, 2014.

[87] E. Zio and L. R. Golea, "Analyzing the topological, electrical and reliability characteristics of a power transmission system for identifying its critical elements," *Reliability Engineering & System Safety,* vol. 101, pp. 67-74, 2012.

[88] W. Li, L. Xie, Z. Deng, and Z. Wang, "False sequential logic attack on SCADA system and its physical impact analysis," *Computers & Security,* vol. 58, pp. 149-159, 2016.

[89] N. Chen and K. L. Tsui, "Condition monitoring and remaining useful life prediction using degradation signals: Revisited," *IIE Transactions,* vol. 45, no. 9, pp. 939-952, 2013.

[90] K. Schneider, C.-C. Liu, and J.-P. Paul, "Assessment of interactions between power and telecommunications infrastructures," *IEEE Transactions on Power Systems,* vol. 21, no. 3, pp. 1123-1130, 2006.

[91] T. J. Mary and P. Rangarajan, "Delay-dependent stability analysis of microgrid with constant and time-varying communication delays," *Electric Power Components and Systems,* vol. 44, no. 13, pp. 1441-1452, 2016.

[92] P. Srikantha and D. Kundur, "A DER attack-mitigation differential game for smart grid security analysis," *IEEE Transactions on Smart Grid,* vol. 7, no. 3, pp. 1476-1485, 2015.

[93] J. Johansson, H. Hassel, and E. Zio, "Reliability and vulnerability analyses of critical infrastructures: Comparing two approaches in the context of power systems," *Reliability Engineering & System Safety,* vol. 120, pp. 27-38, 2013.

[94] A. K. Singh, R. Singh, and B. C. Pal, "Stability analysis of networked control in smart grids," *IEEE Transactions on Smart Grid,* vol. 6, no. 1, pp. 381-390, 2014.

[95] Y. Susuki *et al.*, "A hybrid system approach to the analysis and design of power grid dynamic performance," *Proceedings of the IEEE,* vol. 100, no. 1, pp. 225-239, 2011.

[96] I. Eusgeld, W. Kröger, G. Sansavini, M. Schläpfer, and E. Zio, "The role of network theory and object-oriented modeling within a framework for the vulnerability analysis of critical infrastructures," vol. 94, no. 5, pp. 954-963, 2009.

[97] P. Hines, E. Cotilla-Sanchez, and S. Blumsack, "Do topological models provide good information about vulnerability in electric power networks?," *arXiv preprint arXiv:1002.2268,* 2010.

[98] J. Johansson, S. LaRocca, H. Hassel, and S. Guikema, "Comparing topological performance measures and physical flow models for vulnerability analysis of power

systems," in *International Probabilistic Safety Assessment and Management Conference and the Annual European Safety and Reliability Conference 2012 (PSAM11 ESREL 2012)*, 2012, pp. 6863-6872.

[99] J. Yan, H. He, X. Zhong, and Y. Tang, "Q-learning-based vulnerability analysis of smart grid against sequential topology attacks," *IEEE Transactions on Information Forensics and Security,* vol. 12, no. 1, pp. 200-210, 2016.

[100] J. Bae, S. Lee, Y.-W. Kim, and J.-H. Kim, "Protection strategies against false data injection attacks with uncertain information on electric power grids," 2017.

[101] Y. Han, Y. Wen, C. Guo, and H. Huang, "Incorporating cyber layer failures in composite power system reliability evaluations," *Energies,* vol. 8, no. 9, pp. 9064-9086, 2015.

[102] B. Falahati, Y. Fu, and L. Wu, "Reliability assessment of smart grid considering direct cyber-power interdependencies," *IEEE Transactions on Smart Grid,* vol. 3, no. 3, pp. 1515-1524, 2012.

[103] B. Falahati and Y. Fu, "Reliability assessment of smart grids considering indirect cyber-power interdependencies," *IEEE Transactions on Smart Grid,* vol. 5, no. 4, pp. 1677-1685, 2014.

[104] H. Lei, C. Singh, and A. Sprintson, "Reliability modeling and analysis of IEC 61850 based substation protection systems," *IEEE Transactions on Smart Grid,* vol. 5, no. 5, pp. 2194-2202, 2014.

[105] H. Lei and C. Singh, "Power system reliability evaluation considering cyber-malfunctions in substations," *Electric Power Systems Research,* vol. 129, pp. 160-169, 2015.

[106] M. Eliassi, A. K. Dashtaki, H. Seifi, M.-R. Haghifam, and C. Singh, "Application of Bayesian networks in composite power system reliability assessment and reliability-based analysis," *IET Generation, Transmission & Distribution,* vol. 9, no. 13, pp. 1755-1764, 2015.

[107] I. Tien and A. Der Kiureghian, "Algorithms for Bayesian network modeling and reliability assessment of infrastructure systems," *Reliability Engineering & System Safety,* vol. 156, pp. 134-147, 2016.

[108] M. Esmalifalak, G. Shi, Z. Han, and L. Song, "Bad data injection attack and defense in electricity market using game theory study," *IEEE Transactions on Smart Grid,* vol. 4, no. 1, pp. 160-169, 2013.

[109] A. Delgadillo, J. M. Arroyo, and N. Alguacil, "Analysis of electric grid interdiction with line switching," *IEEE Transactions on Power Systems,* vol. 25, no. 2, pp. 633-641, 2009.

[110] M. S. Rahman, M. A. Mahmud, A. M. T. Oo, and H. R. Pota, "Multi-agent approach for enhancing security of protection schemes in cyber-physical energy systems," *IEEE transactions on industrial informatics,* vol. 13, no. 2, pp. 436-447, 2016.

[111] K. Zhu, M. Chenine, and L. Nordstrom, "ICT architecture impact on wide area monitoring and control systems' reliability," *IEEE transactions on power delivery,* vol. 26, no. 4, pp. 2801-2808, 2011.

[112] X. Sun, Y. Chen, J. Liu, and S. Huang, "A co-simulation platform for smart grid considering interaction between information and power systems," in *ISGT 2014*, 2014: IEEE, pp. 1-6.

[113] H. Georg, S. C. Müller, N. Dorsch, C. Rehtanz, and C. Wietfeld, "INSPIRE: Integrated co-simulation of power and ICT systems for real-time evaluation," in *2013 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2013: IEEE, pp. 576-581.

[114] M. Baran, R. Sreenath, and N. R. Mahajan, "Extending EMTP for simulating agent based distributed applications," in *14th PSCC*, 2002.

[115] Y. Cao, X. Shi, Y. Li, Y. Tan, M. Shahidehpour, and S. Shi, "A simplified co-simulation model for investigating impacts of cyber-contingency on power system operations," *IEEE Transactions on Smart Grid,* vol. 9, no. 5, pp. 4893-4905, 2017.

[116] L. Wu and G. Kaiser, "FARE: A framework for benchmarking reliability of cyber-physical systems," in *IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, Farmingdale, 2013.

[117] P. Wang and D. W. Coit, "Reliability prediction based on degradation modeling for systems with multiple degradation measures," in *Annual Symposium Reliability and Maintainability, 2004 - RAMS*, Los Angeles, 2004.

[118] S. Song , D. W. Coit, Q. Feng , and H. Peng, "Reliability Analysis for Multi-Component Systems Subject to Multiple Dependent Competing Failure Processes," *IEEE Transactions on Reliability* vol. 63, no. 1, pp. 331 - 345, 2014.

[119] A. L. Goel and K. Okumoto, "Time-Dependent Error-Detection Rate Model for Software Reliability and Other Performance Measures," *IEEE Transactions on Reliability,* vol. R-28, no. 3, pp. 206 - 211, 1979.

[120] H. Pham, *System Software Reliability*. Piscataway: Springer, 2006.

[121] D. R. Jeske and X. Zhang, "Some successful approaches to software reliability modeling in industry," *Journal of Systems and Software,* vol. 74, no. 1, pp. 85-99, 2005.

[122] K. Song, I. H. Chang, and H. Pham, "A Software Reliability Model with a Weibull Fault Detection Rate Function Subject to Operating Environments," *Applied Sciences,* vol. 7, no. 10, 2017.

[123] S. R. Welke, B. W. Johnson, and J. H. Aylor, "Reliability modeling of hardware/software systems," *IEEE Transactions on Reliability,* vol. 44, no. 3, pp. 413 - 418, 1995.

[124] R. K. Iyer and P. Velardi, "Hardware-related software errors: measurement and analysis," *IEEE Transactions on Software Engineering,* no. 2, pp. 223-231, 1985.

[125] S. Song, D. W. Coit, Q. Feng, and H. Peng, "Reliability analysis for multi-component systems subject to multiple dependent competing failure processes," *IEEE Transactions on Reliability,* vol. 63, no. 1, pp. 331-345, 2014.

[126] A. Lisnianski, "Extended block diagram method for a multi-state system reliability assessment," *Reliability Engineering & System Safety,* vol. 92, no. 12, pp. 1601-1607, 2007.

[127] M. Li, J. Liu, J. Li, and B. U. Kim, "Bayesian modeling of multi-state hierarchical systems with multi-level information aggregation," *Reliability Engineering & System Safety,* vol. 124, pp. 158-164, 2014.

[128] D. Nykamp. "An introduction to networks." https://mathinsight.org/network_introduction (accessed Aug 6, 2019).

[129] O. R. Theologou and J. G. Carlier, "Factoring and reductions for networks with imperfect vertices," *IEEE Transactions on Reliability,* vol. 40, no. 2, pp. 210-217, 1991.

[130] V. Gaur, O. Yadav, G. Soni, and A. Rathore, "A Review of Metrics, Algorithms and Methodologies for Network Reliability," in *2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2019: IEEE, pp. 1129-1133.

[131] R. Dash, N. Barpanda, P. Tripathy, and C. R. Tripathy, "Network reliability optimization problem of interconnection network under node-edge failure model," *Applied Soft Computing,* vol. 12, no. 8, pp. 2322-2328, 2012.

[132] K. Yang and G. Yang, "Robust reliability design using environmental stress testing," *Quality and reliability engineering international,* pp. 409-416, 1998.

[133] A. F. Shahraki, O. P. Yadav, and H. Liao, "A review on degradation modelling and its engineering applications," *International Journal of Performability Engineering,* vol. 13, no. 3, p. 299, 2017.

[134] A. Forouzandeh Shahraki, O. P. Yadav, and H. Liao, "A Review on Degradation Modelling and Its Engineering Applications," *International Journal of Performability Engineering,* vol. 13, no. 3, pp. 299-314, 2017.

[135] W. Q. Meeker and Y. Hong, "Reliability meets big data: opportunities and challenges," *Quality Engineering,* vol. 26, no. 1, pp. 102-116, 2014.

[136] G. J. Hahn and N. Doganaksoy, *The role of statistics in business and industry*. John Wiley & Sons, 2011.

[137] A. Volponi and B. Wood, "Aircraft Propulsion Health Management," *System Health Management: With Aerospace Applications,* pp. 389-403, 2011.

[138] K. Spurgeon, W. Tang, Q. Wu, Z. Richardson, and G. Moss, "Dissolved gas analysis using evidential reasoning," *IEE Proceedings-Science, Measurement and Technology,* vol. 152, no. 3, pp. 110-117, 2005.

[139] C. C. Ciang, J.-R. Lee, and H.-J. Bang, "Structural health monitoring for a wind turbine system: a review of damage detection methods," *Measurement science and technology,* vol. 19, no. 12, p. 122001, 2008.

[140] I. Antoniadou, G. Manson, N. Dervilis, T. Barszcz, W. Staszewski, and K. Worden, "Condition monitoring of a wind turbine gearbox using the empirical mode decomposition method and outlier analysis," in *Sixth European Workshop on Structural Health Monitoring, Dresden, Germany,* 2012, vol. 3, no. 6.

[141] P. Faulkner, P. Cutter, and A. Owens, "Structural health monitoring systems in difficult environments—offshore wind turbines," in *6th European workshop on structural health monitoring,* 2012, pp. 1-7.

[142] B. Leever and E. Forsythe, "NextFlex: Enabling a Domestic Manufacturing Ecosystem for Flexible Hybrid Electronics," *International Society of coating and Technology,*

[143] S. Peterson, "Power Electronics," ed, 2005.

[144] E. A. Elsayed, "Overview of Reliability Testing," *IEEE Transactions on Reliability,* vol. 61, no. 2, pp. 282 - 291, 2012.

[145] M. Pecht, *Handbook of electronic package design*. CRC Press, 1991.

[146] P. Lall, J. Narangaparambil, A. Abrol, B. Leever, and J. Marsh, "Development of Test Protocols for the Flexible Substrates in Wearable Applications," in *IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm),* San Diego, 2018.

[147] Y. Zhang *et al.*, "Super-stretchy lithium-ion battery based on carbon nanotube fiber," *Journal of Materials Chemistry A,* vol. 2, no. 29, pp. 11054-11059, 2014.

[148] Y. Zhang *et al.*, "Flexible and stretchable lithium-ion batteries and supercapacitors based on electrically conducting carbon nanotube fiber springs," *Angewandte Chemie International Edition,* vol. 53, no. 52, pp. 14564-14568, 2014.

[149] M. M. Hamasha, K. Alzoubi, and S. Lu, "Behavior of sputtered indium–tin–oxide thin film on poly-ethylene terephthalate substrate under stretching," *Journal of Display Technology,* vol. 7, no. 8, pp. 426-433, 2011.

[150] W. Lee, Y. Kim, M. Y. Lee, J. H. Oh, and J. U. Lee, "Highly stretchable fiber transistors with all-stretchable electronic components and graphene hybrid electrodes," *Organic Electronics,* vol. 69, pp. 320-328, 2019, doi: https://doi.org/10.1016/j.orgel.2019.03.056.

[151] C. Wang, W. Zheng, Z. Yue, C. O. Too, and G. G. Wallace, "Buckled, stretchable polypyrrole electrodes for battery applications," *Advanced materials,* vol. 23, no. 31, pp. 3580-3584, 2011.

[152] P. Lall, K. Goyal, B. Leever, and S. Miller, "Reliability of Additively Printed Traces on Polymer Substrates Subjected to Mechanical Stretching," in *2019 18th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm)*, 2019: IEEE, pp. 643-648.

[153] M. Drack, I. Graz, T. Sekitani, T. Someya, M. Kaltenbrunner, and S. Bauer, "An imperceptible plastic electronic wrap," *Advanced materials,* vol. 27, no. 1, pp. 34-40, 2015.

[154] N. Zhang *et al.*, "Biaxially stretchable supercapacitors based on the buckled hybrid fiber electrode array," *Nanoscale,* vol. 7, no. 29, pp. 12492-12497, 2015.

[155] K. J. Kim, J. A. Lee, M. D. Lima, R. H. Baughman, and S. J. Kim, "Highly stretchable hybrid nanomembrane supercapacitors," *RSC Advances,* vol. 6, no. 29, pp. 24756-24759, 2016.

[156] D. E. Leber, B. N. Meek, S. D. Leija, D. G. Wilson, R. L. Chaney, and D. R. Hackler, "Electromechanical reliability testing of flexible hybrid electronics incorporating FleX silicon-on-polymer ICs," in *2016 IEEE Workshop on Microelectronics and Electron Devices (WMED)*, 2016: IEEE, pp. 1-4.

[157] S.-m. Sim, Y. Lee, H.-L. Kang, K.-Y. Shin, S.-H. Lee, and J.-M. Kim, "RF performance of ink-jet printed microstrip lines on rigid and flexible substrates," *Microelectronic Engineering,* vol. 168, pp. 82-88, 2017.

[158] Y. Khan *et al.*, "Flexible Hybrid Electronics: Direct Interfacing of Soft and Hard Electronics for Wearable Health Monitoring," *Advanced Functional Materials,* vol. 26, no. 47, pp. 8764-8775, 2016, doi: 10.1002/adfm.201603763.

[159] Y.-W. Huang and S.-T. Lu, "Development and Reliability of Ultra-Thin Chip on Plastic Bonding for Flexible Liquid Crystal Displays," in *Electronic Components and Technology Conference (ECTC), 2010 Proceedings 60th*, Las Vegas, 2010.

[160] L. Frisk and A. Cumini, "Effect of substrate material and thickness on reliability of ACA bonded flip chip joints," *Soldering & Surface Mount Technology,* vol. 21, no. 3, pp. 16-23, 2009.

[161] K. Saarinen-Pulli, S. Lahokallio, and L. Frisk, "Effects of different anisotropically conductive adhesives on the reliability of UHF RFID tags," *International Journal of Adhesion & Adhesives,* vol. 64, pp. 52-59, 2016.

[162] S. Lahokallio, K. Saarinen-Pulli, and L. Frisk, "Effects of different test profiles of temperature cycling tests on the reliability of RFID tags," *Microelectronics Reliability,* vol. 55, no. 1, pp. 93-100, 2014.

[163] K. Saarinen and L. Frisk, "Effects of different temperature cycling test profiles on the reliability of UHF RFID tags," in *Electronic System-Integration Technology Conference (ESTC), 2012 4th*, Amsterdan, 2012.

[164] K. Saarinen and L. Frisk, "Reliability of UHF RFID Tags in Humid Environments," in *Electronics Packaging Technology Conference (EPTC), 2012 IEEE 14th*, Singapore, 2012.

[165] S.-T. Lu and W.-H. Chen, "Experimental/Numerical Analysis of Thermally Induced Warpage of Ultrathin Chip-on-Flex (UTCOF) Interconnects," *IEEE Transactions on Components and Packaging Technologies,* vol. 33, no. 4, pp. 819 - 829, 2010.

[166] L. Frisk and K. S.-P. Saarinen-Pulli, "Reliability of adhesive joined thinned chips on flexible substrates under humid conditions," *Microelectronics Reliability,* vol. 54, pp. 2058–2063, 2014.

[167] C. Y. Yin, H. Lu, C. Bailey, and Y. C. Chan, "Experimental and modelling analysis on moisture induced failures in flip-chip-on-flex interconnection with anisotropic conductive film," in *Asian Green Electronics*, Shanghai, China, 2005.

[168] P. Palm, J. Maattanen, A. Tuominen, and E. Ristolainen, "Reliability of 80 Mm pitch flip chip attachment on flex," *Microelectronics reliability,* vol. 41, no. 5, pp. 633-638, 2001.

[169] L. Frisk and S. Lahokallio, "Reliability of ICA attached sensor components under prolonged humid conditions," in *Microelectronics Packaging (NordPac)*, Gothenburg, Sweden, 2017: IEEE, pp. 129-134.

[170] X. Cai, B. An, F. Wu, and Y. Wu, "Assembly of flexible RFID tag inlays with anisotropic conductive paste (ACP)," *Circuit World,* vol. 35, no. 4, pp. 40-45, 2009.

[171] S. Saarinen , T. Bjorninen, L. Ukkonen, and L. Frisk, "Reliability Analysis of RFID Tags in Changing Humid Environment," *IEEE Transactions on Components, Packaging and Manufacturing Technology,* vol. 4, no. 1, pp. 77-85, 2013.

[172] M.-U. Hassan, C. Schomburg, C. Harendt, E. Penteker, and J. N. Burghartz, "Assembly and embedding of ultra-thin chips in polymers," in *Eurpoean Microelectronics Packaging Conference (EMPC)*, Grenoble, 2013: IEEE, pp. 1-6.

[173] R. S. Dahiya and S. Gennaro, "Bendable Ultra-Thin Chips on Flexible Foils," *IEEE Sensors Journal,* pp. 4030 - 4037, 2013.

[174] J. N. Burghartz, W. Appel, C. Harendt, H. Rempp, H. Richter, and M. Zimmermann, "Ultra-thin chip technology and applications, a new paradigm in silicon technology," *Solid-State Electronics,* vol. 54, no. 9, pp. 818-829, 2010.

[175] D. A. van den Ende *et al.*, "Mechanical and electrical properties of ultra-thin chips and flexible electronics assemblies during bending," *Microelectronics Reliability,* vol. 54, no. 12, pp. 2860-2870, 2014.

[176] N. Wacker *et al.*, "Stress analysis of ultra-thin silicon chip-on-foil electronic assembly under bending," *Semiconductor Science and Technology,* p. 095007, 2014.

[177] S.-T. Lu and W.-H. Chen, "Reliability and Flexibility of Ultra-Thin Chip-on-Flex (UTCOF) Interconnects With Anisotropic Conductive Adhesive (ACA) Joints," *IEEE Transactions on Advanced Packaging,* vol. 33, no. 3, pp. 702 - 712, 2010.

[178] V. R. Marinov, "The IC in the Flexible Hybrid Electronics Technology: Flexibility and Bend Testing," in *IMAPS 50th Int. Symposium on Microelectronics*, Raleigh, 2017, vol. 2017, pp. 000103-000108.

[179] L. Leppänen, "Bendability of flip-chip attachment on screen printed interconnections. Thesis," Tampere university of Techonology, Tampere, 2016.

[180] M. Poliks, J. Turner, K. Ghose, Z. Jin, M. Garg, and Q. Gui, "A Wearable Flexible Hybrid Electronics ECG Monitor," in *Electronic Components and Technology Conference (ECTC), 2016 IEEE 66th*, Las Vegas, 2016.

[181] C. Landesberger *et al.*, "Novel processing scheme for embedding and interconnection of ultra-thin IC devices in flexible chip foil packages and recurrent bending reliability analysis," in *International Conference on Electronics Packaging (ICEP)*, Sapporo, Japan, 2016.

[182] J.-H. Jeong, J.-H. Kim, and C.-S. Oh, "Quantitative evaluation of bending reliability for a flexible near-field communication tag," *Microelectronics Reliability,* vol. 75, pp. 121-126, 2017.

[183] K. Janeczek, "Reliability analysis of UHF RFID tags under long-term mechanical cycling," *Microelectronics Reliability,* vol. 75, pp. 96-101, 2017.

[184] Y. C. Kim, S. J. Lee, I.-K. Oh, S. Seo, H. Kim, and J.-M. Myoung, "Bending stability of flexible amorphous IGZO thin film transistors with transparent IZO/Ag/IZO oxideemetaleoxide electrodes," *Journal of Alloys and Compounds,* vol. 668, pp. 1108-1114, 2016.

[185] H.-l. Kao, C.-S. Yeh, M.-T. Chen, H.-C. Chiu, and L.-C. Chang, "Characterization and reliability of nMOSFETs on flexible substrates under mechanical strain," *Microelectronics Reliability,* vol. 52, no. 6, pp. 999-1004, 2012.

[186] J. Pan, "A control-chart-based method for solder joint crack detection," *Journal of Microelectronics and Electronic Packaging,* vol. 11, no. 3, pp. 94-103, 2014.

[187] C. W. Park, B. S. Na, J. B. Koo, J.-Y. Oh, N.-M. Park, and S. S. Lee, "Hybrid substrate structure for readily stretchable and highly reliable electronic circuits," in *2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, 2016: IEEE, pp. 1-3.

[188] J. W. Park and B. J. Yum, "Optimal design of accelerated life tests with two stresses," *Naval Research Logistics (NRL),* vol. 43, no. 6, pp. 863-884, 1996.

[189] T.-R. Tsai, Y. Lio, and N. Jiang, "Optimal decisions on the accelerated degradation test plan under the Wiener process," *Quality Technology & Quantitative Management,* vol. 11, no. 4, pp. 461-470, 2014.

[190] W. Nelson, *Accelerated Testing Statistical Models, Test Plans, and Data Analysis*. New York: Wiley, 2004.

[191] S. Dusmez and B. Akin, "An accelerated thermal aging platform to monitor fault precursor on-state resistance," in *2015 IEEE International Electric Machines & Drives Conference (IEMDC)*, 2015: IEEE, pp. 1352-1358.

[192] N. Gebraeel, "Sensory-updated residual life distributions for components with exponential degradation patterns," *IEEE Transactions on Automation Science and Engineering,* vol. 3, no. 4, pp. 382-393, 2006.

[193] F. Dia *et al.*, "Model Associated with the study of the degradation Based on the Accelerated Test: A Literature Review," *Open Journal of Applied Sciences,* vol. 6, no. 01, p. 49, 2016.

[194] M. D. Kempe, "Modeling of rates of moisture ingress into photovoltaic modules," *Solar Energy Materials and Solar Cells,* vol. 90, no. 16, pp. 2720-2738, 2006.

[195] J. Kapur, K. Proost, and C. A. Smith, "Determination of moisture ingress through various encapsulants in glass/glass laminates," in *2009 34th IEEE Photovoltaic Specialists Conference (PVSC),* 2009: IEEE, pp. 001210-001214.

[196] D. Wu, J. Zhu, T. R. Betts, and R. Gottschalg, "Degradation of interfacial adhesion strength within photovoltaic mini-modules during damp-heat exposure," *Progress in Photovoltaics: Research and Applications,* vol. 22, no. 7, pp. 796-809, 2014.

[197] K. Kokko, "Reliability of ACA Joints with Conformal Coatings in Harsh Environments," 2010.

[198] J. C. Pinheiro and D. M. Bates, "Statistics and computing. Mixed-effects models in S and S-PLUS," ed: Springer-Verlag, New York, NY, 2000.

[199] J. Pinheiro, D. Bates, S. DebRoy, D. Sarkar, S. Heisterkamp, and B. Van Willigen, "Package ‘nlme’. Linear and Nonlinear Mixed Effects Models," 2019.

[200] M. Nagode and M. Fajdiga, "On a new method for prediction of the scatter of loading spectra," *International Journal of Fatigue,* vol. 20, no. 4, pp. 271-277, 1998.

[201] L. Min, X. Xiaofei, and Y. Qing-Xiong, "Cumulative fatigue damage dynamic interference statistical model," *International Journal of Fatigue,* vol. 17, no. 8, pp. 559-566, 1995.

[202] J. R. Benjamin and C. A. Cornell, *Probability, statistics, and decision for civil engineers*. Courier Corporation, 2014.

[203] Y. Liu and S. Mahadevan, "Stochastic fatigue damage modeling under variable amplitude loading," *International Journal of Fatigue,* vol. 29, no. 6, pp. 1149-1161, 2007.

[204] W. Hwang and K. Han, "Cumulative damage models and multi-stress fatigue life prediction," *Journal of composite materials,* vol. 20, no. 2, pp. 125-153, 1986.

[205] ZVEI, *Handbook for Robustness Validation of Semiconductor Devices in Automotive Applications*. Frankfurt: ZVEI - Zentralverband Elektrotechnik- und Elektronikindustrie e. V., 2015.

[206] L. Escobar and W. Meeker, "A Review of Accelerated Test Models," *Statistical Science,* vol. 21, pp. 552-577, 2006.

[207] C. Park and W. J. Padgett, "Stochastic degradation models with several accelerating variables," *IEEE Transactions on Reliability,* vol. 55, no. 2, pp. 379-390, 2006.

[208] Y. Wang, C. Zhang, S. Zhang, X. Chen, and Y. Tan, "Optimal design of constant stress accelerated degradation test plan with multiple stresses and multiple degradation measures," *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability,* vol. 229, no. 1, pp. 83-93, 2015.

[209] A. C. Rencher and C. W. F, *Methods of multivariate analysis*. Wiley Series in Probability and Statistics, 2012.

[210] V. Rathod, O. P. Yadav, A. Rathore, and R. Jain, "Probabilistic modeling of fatigue damage accumulation for reliability prediction," *International Journal of Quality, Statistics, and Reliability,* vol. 2011, 2011.

[211] J. G. Eisenhauer, "Regression through the origin," *Teaching statistics,* vol. 25, no. 3, pp. 76-80, 2003.

[212] H. Hecht and M. Hecht, "Software reliability in the system context," *IEEE Transactions on Software Engineering,* no. 1, pp. 51-58, 1986.

[213] M. A. Friedman, P. Tran, and P. L. Goddard, "Reliability techniques for combined hardware and software systems," ROME LAB ROME NY, 1992.

[214] R. K. Iyer, D. J. Rossetti, and M.-C. Hsueh, "Measurement and modeling of computer reliability as affected by system activity," *ACM Transactions on Computer Systems (TOCS),* vol. 4, no. 3, pp. 214-237, 1986.

[215] J. Gray, "A census of Tandem system availability between 1985 and 1990," *IEEE Transactions on reliability,* vol. 39, no. 4, pp. 409-418, 1990.

[216] T.-T. Lin and D. P. Siewiorek, "Error log analysis: statistical modeling and heuristic trend analysis," *IEEE Transactions on reliability,* vol. 39, no. 4, pp. 419-432, 1990.

[217] B. Schroeder and G. A. Gibson, "Disk failures in the real world: What does an MTTF of 1, 000, 000 hours mean to you?," in *FAST*, 2007, vol. 7, no. 1, pp. 1-16.

[218] N. El-Sayed and B. Schroeder, "Reading between the lines of failure logs: Understanding how HPC systems fail," in *2013 43rd annual IEEE/IFIP international conference on dependable systems and networks (DSN)*, 2013: IEEE, pp. 1-12.

[219] D. Tang and R. K. Iyer, "Analysis of the VAX/VMS error logs in multicomputer environments-a case study of software dependability," in *[1992] Proceedings Third International Symposium on Software Reliability Engineering*, 1992: IEEE, pp. 216-226.

[220] M. R. Lyu, *Handbook of Software Reliability Engineering*. IEEE Computer Society Press, 1997.

[221] M. R. Bussieck and A. Meeraus, "General algebraic modeling system (GAMS)," in *Modeling languages in mathematical optimization*: Springer, 2004, pp. 137-157.

[222] G. D. Corp. "Stochastic Programming." https://www.gams.com/latest/docs/UG_EMP_SP.html (accessed 10/16, 2019).

[223] M. Ohba and S. Yamada, "S-shaped software reliability growth models," in *International Colloquium on Reliability and Maintainability, 4 th, Tregastel, France*, 1984, pp. 430-436.

[224] Los Alamos National Laboratory. *Ultrascale Systems Research Center (USRC) Data Sources*,

[225] T. Zhang, M. Xie, L. C. Tang, and S. H. Ng, "Reliability and modeling of systems integrated with firmware and hardware," *International Journal of Reliability, Quality and Safety Engineering*, vol. 12, no. 03, pp. 227-239, 2005.

[226] S. Kumar and K. Li, "Performance impact of using ESP to implement VMMC firmware," in *Workshop on Novel Uses of System Area Networks (SAN-1)*, 2002: Citeseer.

[227] IEEE, "Standard Glossary of Software Engineering Terminology (IEEE Std. 610.12-1990)," ed: IEEE New York, EUA, 1990.

[228] D. S. Roy, C. Murthy, and D. K. Mohanta, "Reliability analysis of phasor measurement unit incorporating hardware and software interaction failures," *IET Generation, Transmission & Distribution*, vol. 9, no. 2, pp. 164-171, 2014.

[229] G. M. Siddesh, G. C. Deka, K. G. Srinivasa, and L. M. Patnaik, *Cyber-Physical Systems: A Computational Perspective*. CRC Press, 2015.

[230] P. Chandrasekar and T. Sangeetha, "Smart shopping cart with automatic billing system through RFID and ZigBee," in *International Conference on Information Communication and Embedded Systems (ICICES2014)*, 2014: IEEE, pp. 1-4.

[231] L. Bain, *Statistical analysis of reliability and life-testing models: theory and methods*. Routledge, 2017.

[232] G. Salvendy, *Handbook of industrial engineering: technology and operations management*. John Wiley & Sons, 2001.

[233] A. Shapiro and A. Nemirovski, "On complexity of stochastic programming problems," in *Continuous optimization*: Springer, 2005, pp. 111-146.

[234] H. Pham, "Recent studies in software reliability engineering," in *Handbook of Reliability Engineering*: Springer, 2003, pp. 285-302.

[235] H. Pham, *Software Reliability*. Piscataway: Springer, 2000.

[236] O. Mersmann, H. Trautmann, D. Steuer, B. Bornkamp, and M. O. Mersmann, "Package ‘truncnorm’," Technical report. URL https://github. com/olafmersmann/truncnorm, 2018.

[237] H. Bengtsson *et al.*, "Package ‘matrixStats’," 2019.

[238] R. V. Hogg, J. W. McKean, and A. T. Craig, *Introduction to Mathematical Statistics*, 7th ed. Boston, 2013.

[239] I. Gertsbakh and Y. Shpungin, *Network reliability and resilience*. Springer Science & Business Media, 2011.

[240] L. d. F. Costa, F. A. Rodrigues, G. Travieso, and P. R. Villas Boas, "Characterization of complex networks: A survey of measurements," *Advances in physics,* vol. 56, no. 1, pp. 167-242, 2007.

[241] M. R. Garey, "David S. Johnson Computers and Intractability," *AGuide to the Theory of NP-Completeness WH Freeman and CompanyNew York,* 1979.

[242] R. Jan, "«Design of reliable networks), Computers and Operations Research. Vol. 20," 1993.

[243] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon,* vol. 4, no. 11, p. e00938, 2018.

[244] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, 2017: IEEE, pp. 1-6.

[245] R. Van Slyke and H. Frank, "Network reliability analysis: Part I," *Networks,* vol. 1, no. 3, pp. 279-290, 1971.

[246] R. Wilkov, "Analysis and design of reliable computer networks," *IEEE Transactions on Communications,* vol. 20, no. 3, pp. 660-678, 1972.

[247] M. J. Zaki, W. Meira Jr, and W. Meira, *Data mining and analysis: fundamental concepts and algorithms*. Cambridge University Press, 2014.

[248] K. T. Chui and C.-w. Shen, "Tolerance analysis in scale-free social networks with varying degree exponents," *Library Hi Tech,* 2019.

[249] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Reviews of modern physics,* vol. 74, no. 1, p. 47, 2002.

[250] P. Crucitti, V. Latora, M. Marchiori, and A. Rapisarda, "Efficiency of scale-free networks: error and attack tolerance," *Physica A: Statistical Mechanics and its Applications,* vol. 320, pp. 622-642, 2003.

[251] M. Ball and R. M. Van Slyke, "Backtracking algorithms for network reliability analysis," in *Annals of discrete Mathematics,* vol. 1: Elsevier, 1977, pp. 49-64.

[252] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research,* vol. 15, no. 1, pp. 1929-1958, 2014.

[253] J. M. Twomey and A. E. Smith, "Bias and variance of validation methods for function approximation neural networks under conditions of sparse data," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* vol. 28, no. 3, pp. 417-430, 1998.

[254] J. Arunnehru, A. Kumar, and J. Verma, "Early Prediction of Brain Tumor Classification Using Convolution Neural Networks," in *International Conference on Computational Intelligence, Security and Internet of Things,* 2019: Springer, pp. 16-25.

[255] A. Vedaldi and K. Lenc, "Matconvnet: Convolutional neural networks for matlab," in *Proceedings of the 23rd ACM international conference on Multimedia*, 2015, pp. 689-692.

[256] D. Masters and C. Luschi, "Revisiting small batch training for deep neural networks," *arXiv preprint arXiv:1804.07612*, 2018.

[257] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural networks*, vol. 12, no. 1, pp. 145-151, 1999.

[258] P. Khumprom and N. Yodo, "A data-driven predictive prognostic model for lithium-ion batteries based on a deep learning algorithm," *Energies*, vol. 12, no. 4, p. 660, 2019.

[259] T. Jin and D. W. Coit, "Approximating network reliability estimates using linear and quadratic unreliability of minimal cuts," *Reliability Engineering & System Safety*, vol. 82, no. 1, pp. 41-48, 2003.

[260] A. Narayanan, M. Chandramohan, R. Venkatesan, L. Chen, Y. Liu, and S. Jaiswal, "graph2vec: Learning distributed representations of graphs," *arXiv preprint arXiv:1707.05005*, 2017.

[261] F. Gao, G. Wolf, and M. Hirn, "Geometric scattering for graph data analysis," in *International Conference on Machine Learning*, 2019, pp. 2122-2131.

[262] H. Chen and H. Koga, "GL2vec: Graph Embedding Enriched by Line Graphs with Edge Features," in *International Conference on Neural Information Processing*, 2019: Springer, pp. 3-14.

[263] S. Verma and Z.-L. Zhang, "Hunt for the unique, stable, sparse and fast feature learning on graphs," in *Advances in Neural Information Processing Systems*, 2017, pp. 88-98.

[264] A. Tsitsulin, D. Mottin, P. Karras, A. Bronstein, and E. Müller, "Netlsd: hearing the shape of a graph," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2347-2356.

[265] N. de Lara and E. Pineau, "A simple baseline algorithm for graph classification," *arXiv preprint arXiv:1810.09155*, 2018.

[266] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[267] J. Brownlee, *Better Deep Learning: Train Faster, Reduce Overfitting, and Make Better Predictions*. Machine Learning Mastery, 2018.

[268] H. Choi, S. Subramaniam, and H.-A. Choi, "Loopback recovery from double-link failures in optical mesh networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 6, pp. 1119-1130, 2004.

[269] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The internet topology zoo," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 9, pp. 1765-1775, 2011.

[270] H. Cancela, P. L'Ecuyer, G. Rubino, and B. Tuffin, "Combination of conditional Monte Carlo and approximate zero-variance importance sampling for network reliability estimation," in *Proceedings of the 2010 Winter Simulation Conference*, 2010: IEEE, pp. 1263-1274.

[271] G. Zhao, G. Zhang, Q. Ge, and X. Liu, "Research advances in fault diagnosis and prognostic based on deep learning," in *2016 Prognostics and System Health Management Conference (PHM-Chengdu)*, 2016: IEEE, pp. 1-6.

[272] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11-26, 2017.

[273] L. Guo, N. Li, F. Jia, Y. Lei, and J. Lin, "A recurrent neural network based health indicator for remaining useful life prediction of bearings," *Neurocomputing*, vol. 240, pp. 98-109, 2017.

[274] J. Liu, A. Saxena, K. Goebel, B. Saha, and W. Wang, "An adaptive recurrent neural network for remaining useful life prediction of lithium-ion batteries," NATIONAL AERONAUTICS AND SPACE ADMINISTRATION MOFFETT FIELD CA AMES RESEARCH …, 2010.

[275] H. Wang, Z. Yang, Q. Yu, T. Hong, and X. Lin, "Online reliability time series prediction via convolutional neural network and long short term memory for service-oriented systems," *Knowledge-Based Systems,* vol. 159, pp. 132-147, 2018.

[276] J. Li, P. He, J. Zhu, and M. R. Lyu, "Software defect prediction via convolutional neural network," in *2017 IEEE International Conference on Software Quality, Reliability and Security (QRS)*, 2017: IEEE, pp. 318-328.

[277] N. Akai, L. Y. Morales, and H. Murase, "Simultaneous pose and reliability estimation using convolutional neural network and Rao–Blackwellized particle filter," *Advanced Robotics,* vol. 32, no. 17, pp. 930-944, 2018.

[278] Q. Wang, B. Zhao, H. Ma, J. Chang, and G. Mao, "A method for rapidly evaluating reliability and predicting remaining useful life using two-dimensional convolutional neural network with signal conversion," *Journal of Mechanical Science and Technology,* vol. 33, no. 6, pp. 2561-2571, 2019.

[279] L. Ren, L. Zhao, S. Hong, S. Zhao, H. Wang, and L. Zhang, "Remaining useful life prediction for lithium-ion battery: A deep learning approach," *IEEE Access,* vol. 6, pp. 50587-50598, 2018.

[280] L. Ren, Y. Sun, J. Cui, and L. Zhang, "Bearing remaining useful life prediction based on deep autoencoder and deep neural networks," *Journal of Manufacturing Systems,* vol. 48, pp. 71-77, 2018.

[281] S. Knight. "The Internet Topology Zoo." The University of Adelaide. http://www.topology-zoo.org (accessed 3/16, 2021).

[282] F. Pascual, W. Q. Meeker, and L. A. Escobar, "Accelerated life test models and data analysis," *Handbook of Engineering Statistics (H. Pham, ed.) Chapter,* vol. 22, 2006.

[283] J. C. Pinheiro and D. M. Bates, "Approximations to the Log-Likelihood Function in the Nonlinear Mixed-Effects Model," *Journal of Computational and Graphical Statistics,* vol. 4, no. 1, pp. 12-35, 1995.

[284] ITU, *Optical Fibres, cables and systems*. 2010.

[285] M. W. Beranek and A. R. Avak, "Improving avionics fiber optic network reliability and maintainability via built-in test," in *2006 ieee/aiaa 25TH Digital Avionics Systems Conference*, 2006: IEEE, pp. 1-5.

[286] M. J. Matthewson, "Optical fiber reliability models," in *Fiber Optics Reliability and Testing: A Critical Review,* 1993, vol. 10272: International Society for Optics and Photonics, p. 1027203.

[287] G. M. Bubel and M. J. Matthewson, "Optical fiber reliability implications of uncertainty in the fatigue crack growth model," *Optical Engineering,* vol. 30, no. 6, pp. 737-745, 1991.

[288] S. Flint, "Failure rates for fiber optic assemblies," IIT RESEARCH INST CHICAGO IL, 1980.

[289] N. E. Dowling, *Mechanical behavior of materials: engineering methods for deformation, fracture, and fatigue*. Pearson, 2012.

[290] M. Li and W. Q. Meeker, "Application of Bayesian methods in reliability data analyses," *Journal of Quality Technology,* vol. 46, no. 1, pp. 1-23, 2014.

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

[292] A. Gelman, J. B. Carlin, and H. S. Stern, *Bayesian Data Analysis*, 3rd ed. (Texts in Statistical Science). Boca Raton: CRC Press, 2013.

[293] Y. Zhan, Q. Yang, H. Wu, J. Lei, and P. Liang, "Degradation of beam quality and depolarization of the laser beam in a step-index multimode optical fiber," *Optik,* vol. 120, no. 12, pp. 585-590, 2009.

[294] G. P. Agrawal, *Fiber-optic communication systems*. John Wiley & Sons, 2012.

[295] I. Ntzoufras, *Bayesian modeling using WinBUGS*. John Wiley & Sons, 2011.

[296] D. J. Spiegelhalter, A. Thomas, N. G. Best, W. Gilks, and D. Lunn, "BUGS: Bayesian inference using Gibbs sampling," *Version 0.5,(version ii) http://www. mrc-bsu. cam. ac. uk/bugs,* vol. 19, 1996.

[297] J. Liu, D. J. Nordman, and W. Q. Meeker, "The number of MCMC draws needed to compute Bayesian credible bounds," *The American Statistician,* vol. 70, no. 3, pp. 275-284, 2016.

[298] R. Billinton and W. Li, *Reliability assessment of electric power systems using Monte Carlo methods*. Springer Science & Business Media, 2013.

[299] R. Zhou, J. Liu, S. Kumar, and D. P Palomar, "Robust Factor Analysis Parameter Estimation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* p. 3, 2020.