A Deep Neural Network and Bayesian Method based Framework for All-Terminal Network Reliability Estimation Considering Degradation

Alex Davila-Frias[1,2], Nita Yodo[1], Trung Le[1], and Om Prakash Yadav[1,*]
[1]Department of Industrial & Manufacturing Engineering, North Dakota State University.
[2]Facultad de Ciencias Administrativas, Escuela Politécnica Nacional, Ecuador
Email: om.yadav@ndsu.edu
Department of Industrial & Manufacturing Engineering, North Dakota State University, NDSU Dept.2485, P.O. Box6050, Fargo, ND 58108-6050, USA

Abstract

Most research on network reliability has considered links to have a binary state, i.e., functioning or failed, whereas nodes are considered flawless. In a more realistic scenario, both links and nodes might fail or may exhibit degradation behavior before failing. This study develops a framework to estimate the all-terminal reliability of a network that considers the degradation and probability of failure of all nodes and links in a network. Unlike previous works on network reliability that considered constant reliability for links, this paper considers the reliability of links, nodes, and the network as functions of time. In the proposed framework, the Bayesian methods (BM) are employed to estimate the 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 MC algorithm can estimate 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 proposed framework can provide not only reliability point estimates but also credible intervals. Finally, we take advantage of Bayesian methods to integrate new data into the framework as they become available. The framework uses the new data to refine and further update the degradation model parameters and the prediction of the reliability of links, nodes,

and the network. The proposed methodology has been demonstrated with the real-world network topology Ion (125 nodes, 150 links) with actual degradation data.

Keywords: All-terminal network reliability, Bayesian methods, credible intervals, degradation, Monte Carlo, Deep neural networks.

List of abbreviations

ANN	Artificial neural network
BM	Bayesian methods
CDF	Cumulative distribution function
CNN	Convolutional neural network
DNN	Deep neural network
MC	Monte Carlo
MLE	Maximum likelihood estimate
PDF	Probability density function
PMF	Probability mass function
RMSE	Root mean square error
RNN	Recurrent neural network
RUL	Remaining useful life

List of symbols

$\mathrm{f}(\cdot)$	Probability density function
$\Pr(\cdot)$	Probability function
ν_0	Degrees of freedom
$\Phi_{\mathrm{nor}}(\cdot)$	Standardized normal CDF
D_f	Threshold level
D_{ijk}	Actual degradation y_{ijk} for sample i, at time t_j under stress level k
$F(\cdot)$	Cumulative distribution function
R_{links}	Reliability of links
R_{net}	Network reliability
R_{nodes}	Reliability of nodes
b_i	i^{th} slope in the linear degradation model
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
$\boldsymbol{S_0}$	Inverse of scale matrix
β_i	i^{th} degradation model parameter
$\varphi_{\mathrm{nor}}(\cdot)$	Standardized normal PDF
$\varphi_{\mathrm{nor}}(\cdot)$	Standardized normal PDF
M	Number of replications
$R(\cdot)$	Reliability function

t	time
$\boldsymbol{T}$	Precision matrix
$\boldsymbol{\Sigma}$	Covariance matrix
μ	Mean
σ	Standard deviation
τ	Square-root transformation for time
ϵ	Residual deviation
$\boldsymbol{\theta}$	Vector of unknown parameters
$\boldsymbol{\mu}$	Mean vector

1. Introduction

Networks are commonly used to represent interconnected infrastructure systems such as computer networks, piping systems, or power supply systems [1-3]. Therefore, reliability assessment of these critical networks is imperative. Moreover, reliability assessment of the networks and their components is critical for the users as well as for the producers, especially in logistical decision making such as preventive maintenance, warranty policy, and spare parts management. A network can be defined as a set of items (nodes or vertices) connected by edges or links [4]. Graphical models allow visualizing the interdependencies of the components in a system. Nodes characterize components and junctions of the system, and links represent the connections. For example, busbars in power systems or switches in telecommunication systems are modeled by nodes, whereas links 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 [5-7].

Regardless of the number of nodes, links, or their interconnection, network reliability has several definitions. Most of them are associated with connectivity [8]. Three popular measures are all-terminal, two-terminal, and *k*-terminal [9]. All-terminal reliability is the probability that every node can communicate with every other node in the network, i.e., the network forms at least a minimum spanning tree [10]. The two-terminal reliability problem requires that a pair of specified nodes, e.g., source (s) and terminal (t), 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 slighlty

simpler than the all-terminal reliability problem [1], it might provide a dubious view of the system's network [11, 12]. Advanced network reliability techniques have been oriented to the *k*-terminal and all-terminal reliability [13-17]. This paper focuses on the all-terminal reliability, as it has practical applications, especially in communications networks [14, 18-21]. It requires all the nodes to be connected to each other without possible failures, providing a holistic reliability measure for the network.

In most previous works on network reliability, binary states have been commonly assumed for links, and nodes have been considered perfect [1, 14, 17, 22, 23]. Even considering link failures only, the problem is complex and NP-hard [1, 14]. Traditional network reliability methods include exact NP-hard methods [1, 8, 9, 24] or approximated methods. Among approximated methods, there are several methods such as graph reduction [1, 9], cut-set and tie-set approximations [1], Monte Carlo (MC) [25-29], and bounds [1, 30, 31]. More recently, modern approaches based on percolation theory [32], All-Pairs Homogeneity-Arc [33], matrix-exponential [34], minimal cuts for demand (d-MC) [35], binary-addition tree [36, 37], among others, have been applied for network reliability as well.

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 the reliability estimation of networks [38].

ANNs have been usually trained with the network topology and link reliability as inputs and with the target network reliability as desired output [14, 17, 23]. For example, Srivaree- Ratana et al. [14] 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 [1, 10, 39]) as inputs, and the exact network reliability as the target. More recently, Altiparmak et al. [17] 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. [23] proposed a method based on ANNs to maximize the reliability of fully connected networks subjected to some predefined total cost. Traditional ANNs have

evolved to deep learning (DL) approaches [40, 41], such as deep neural networks (DNNs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs). These advanced DL methods have been utilized in the reliability estimation problem. For instance, RNNs have been successfully applied to predict the health and remaining useful life of bearings [42], li-ion batteries [43]. Also, CNNs have been applied to evaluate online services reliability [44], software reliability [45], a robot's pose and reliability [46], rotating machinery reliability [47], and recently, network reliability [48]. Similarly, a DNN has been used for health prognostic of li-ion batteries [49, 50] and RUL of bearing [51]. 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 kinds 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. Research in network reliability estimation considering imperfect nodes is still scarce, i.e., [24, 52, 53]. 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 article is to integrate the concepts of component reliability based on degradation data and network reliability by modeling components as nodes and links that degrade. This paper provides a sophisticated framework to estimate the all-terminal network reliability as an indicator of the overall health condition of the network. The proposed framework utilizes 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 Bayesian methods (BM), MC simulation, and DNNs. Although the complexity of a given network could be first reduced by applying series-parallel laws, in our approach, we avoid this previous step because not all networks represented as series-parallel (sp) are reducible [54, 55] and because even if a network is sp-reducible, it will add extra calculation time and steps. Moreover, sp-reduction may not be convenient since

we will be considering time-varying reliability values for links and nodes. The proposed approach contemplates three main steps. First, BM with low-information prior distributions is proposed to estimate the degradation parameters and 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 to estimate network reliability function for given links and nodes reliability functions. The proposed MC algorithm can provide good estimates of the network reliability for given (fixed) nodes and links reliability values. Nevertheless, even this algorithm might not be practical for real-time applications. Therefore, to speed up the calculation, a DNN model is designed and integrated into the framework.

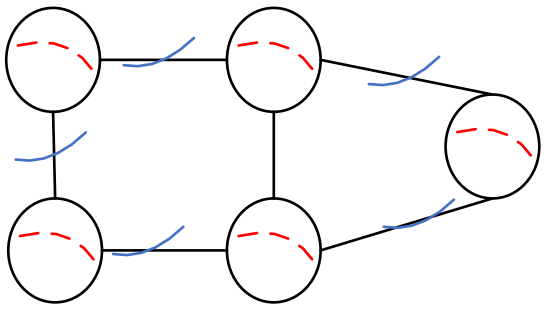
Finally, the DNN model is trained for a range of links and nodes reliability values to learn the all-terminal network reliability calculated with the MC method. In addition, the framework allows the incorporation of new data, as they become available, to update the reliability predictions of links, nodes, and the network. The proposed framework provides point estimates and credible Bayesian intervals for the reliability functions of links, nodes, and all-terminal network reliability functions. To demonstrate the applicability of the proposed approach, the real-world network topology Ion (125 nodes, 150 links, New York, USA) [56, 57] was analyzed using the proposed framework. Real degradation data were considered for nodes [58]. Similarly, the data were simulated based on real degradation data for links [59]. In summary, This work presents a comprehensive approach to evaluate the all-terminal reliability of networked systems. Moreover, we have improved the network reliability assessment framework by considering the time-dependent degradation behavior of both nodes and links in the system. While individual elements of this paper are not new, the integration of these elements makes the proposed method innovative, providing a balance between academic material and practical application.

In the rest of the article, to avoid confusion, the *DNN* acronym will be used to refer to an artificial deep neural network, whereas the term *network* will be employed for the network whose reliability estimation will be performed. The remainder of this article is organized as follows: Section 2 provides a detailed discussion on the proposed methodology, which comprises links and nodes degradation models, MC

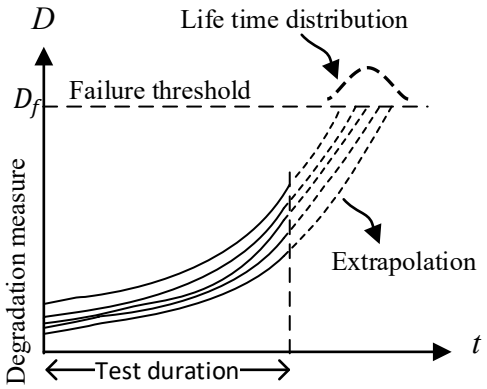
method, DNN model, and Bayesian updating. A case study of network reliability estimation is presented in Section 3. Finally, Section 4 concludes the work and highlights future research opportunities.

2. Proposed Methodology

This section presents the proposed framework for the all-terminal reliability function estimation of a network considering the 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 $G(N, L, p_L(t), p_N(t))$, where N is the set of nodes, L is the set of links, $p_L(t)$ is the reliability of the link, and $p_N(t)$ is the reliability of the nodes. 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 the degradation data of the links and nodes, respectively. This paper 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 1 (a), whereas a general degradation path for a component (link or noded) is represented in Figure 1 (b). 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. In Figure 1 (a), degradation patterns for links are represented by solid blue curves, whereas degradation for nodes is symbolized by the red dashed curves.



(a)



(b)

Figure 1. (a) Representation of a network with degradation in links and nodes. (b) General degradation path for a component

The proposed approach makes the following assumptions for the network:

1. The links and nodes' failure probabilities are independent and equal at a given point of time.
2. The network has non-redundant bi-directional links, e.g., communication or transportation networks. Therefore, a network is modeled with an undirected weighted simple graph.
3. Links and nodes have a performance variable that degrades with time.

The proposed framework is broadly composed of 1) The BM approach for links and nodes degradation models for reliability evaluation. 2) MC 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 the MC method. 4) The Bayesian updating of parameters and network reliability. The four components of the framework are presented in Sections 2.1 to 2.4 and summarized in Section 2.5.

2.1. Links and nodes degradation models for reliability evaluation

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

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

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 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 $\boldsymbol{\Sigma_\beta}$ [60]. 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 $\boldsymbol{\theta_\beta} = \left(\boldsymbol{\mu_\beta}, \boldsymbol{\Sigma_\beta}\right)$ denote the overall population/process parameters.

The likelihood for the random-parameter degradation model is given as [60]:

$$L\left(\boldsymbol{\theta_\beta}, \sigma_\epsilon | DATA\right) = \prod_{i=1}^{n} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \left[\prod_{j=1}^{m_i} \frac{1}{\sigma_\epsilon} \varphi_{\text{nor}}\left(\zeta_{ij}\right) \right] \times f_{\boldsymbol{\beta}}\left(\beta_{1i}, \dots, \beta_{ki}; \boldsymbol{\theta_\beta}\right) d\beta_{1i}, \dots, \beta_{ki} \tag{2}$$

where $\zeta_{ij} = \left[y_{ij} - D(t_{ij}, \beta_{1i}, \dots, \beta_{ki})\right]/\sigma_\epsilon$, $\varphi_{\text{nor}}\left(\zeta_{ij}\right)$ is the standardized normal PDF, and $f_{\boldsymbol{\beta}}\left(\beta_{1i}, \dots, \beta_{ki}; \boldsymbol{\theta_\beta}\right)$ is the multivariate normal distribution density function. The evaluation of equation (2) 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 (2) with respect to $\left(\boldsymbol{\mu_\beta}, \boldsymbol{\Sigma_\beta}, \sigma_\epsilon\right)$ directly can be extremely difficult, although there are some methods [61] and software packages, e.g., 'nmle' [62], 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 the 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:

$$F(t) = \Pr(T \le t) = F\left(t, \boldsymbol{\theta_\beta}\right) = \Pr\left[D(t, \beta_1, \dots, \beta_k) \ge D_f\right] \tag{3}$$

For most practical cases, where $D(t)$ is nonlinear and $\beta_1, \dots, \beta_k$ are random parameters, there is no closed-form expression for $F(t)$, and it has to be evaluated by methods such as numerical integration or MC [60].

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 2.3.

The degradation patterns (e.g., linear, convex, or concave), and its 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 paper is to provide a generic approach to estimate the reliability of a network considering the 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 (2.1.1 and 2.1.2) for links and nodes separately, as well as the expressions to evaluate the corresponding reliability functions $p_L(t)$ and $p_N(t)$. Despite the assumed models as an example only, our proposed approach is generic and can be used with different degradation data/models, provided degradation data and a degradation model properly describing the specific degradation processes are supplied. Although, Sections 2.1.1 and 2.1.2 present specific nonlinear and linear degradation profiles for links and nodes, respectively, different degradation profiles can be considered depending on the actual network and its components.

2.1.1. Links degradation modeling

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 [63, 64]. For instance, when the fiber is exposed to sustained stress, degradation occurs as crack growth [65, 66]. Moreover, cracks are a kind of failure mechanism that leads to the degradation of light transmission capabilities [67]. Hence, a crack growth model is assumed for the degradation of links in this section. The model explained in this section is based on Ref. [60].

Let $a(t)$ be the size of a crack at time t. By the Paris-rule model [68], we have:

$$\frac{da(t)}{dt} = C \times [\Delta K(a)]^m \tag{4}$$

where, C and m are material properties. Typical values of m range from 2 to 4.5. For example, for aluminum 7075-T6, m is equal to 2.836. For this material, the dimensions for C are

$[(cycles)(Pa)^{2.836}(m)^{0.418}]^{-1}$. Different powers of m lead to different powers of the dimensional units [69]. $\Delta K(a)$ is the stress intensity range function. Considering a small crack, $\Delta K(a) = Stress\sqrt{\pi a}$. The solution to the differential equation (4) is:

$$a(t) = \begin{cases} \left[\left(a(0)\right)^{1-\frac{m}{2}} + \left(1-\frac{m}{2}\right) \times C \times \left(Stress\sqrt{\pi}\right)^{m} \times t\right]^{\frac{2}{2-m}}, & m \neq 2 \\ a(0) \times \exp\left[C \times \left(Stress\sqrt{\pi}\right)^{2} \times t\right] & m = 2 \end{cases} \quad (5)$$

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:

$$y_{ij} = a\left(t_{ij}, a(0), \beta_{1i}, \beta_{2i}\right) + \epsilon_{ij} \quad (6)$$

To estimate the parameters, we propose to use BM instead of maximizing equation (2) with respect to $\left(\boldsymbol{\mu_\beta}, \boldsymbol{\Sigma_\beta}, \sigma_\epsilon\right)$ or employing software such as 'nmle' [62] R package to calculate the MLE estimates. Bayesian estimation is a promising alternative to maximum likelihood (ML) and has been 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 [70]. In addition, BM provides not only point estimates but also credible intervals, which can be derived from MCMC draws [70]. Credible intervals are commonly used to describe the Bayesian analog to non-Bayesian confidence intervals [70]. One benefit of BM is that prior information, if available, can be incorporated into the analysis [71, 72], providing improvements in precision or cost savings in testing. 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 2.1.3.

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 :

$$F_L(t) = \int_{-\infty}^{\infty} \Phi_{\text{nor}}\left[-\frac{g(D_f, t, \beta_1) - \mu_{\beta_2|\beta_1}}{\sigma_{\beta_2|\beta_1}}\right]\frac{1}{\sigma_{\beta_1}}\,\varphi_{\text{nor}}\left(\frac{\beta_1 - \mu_{\beta_1}}{\sigma_{\beta_1}}\right)d\beta_1 \tag{7}$$

where, $\Phi_{\text{nor}}(\cdot)$ is the standardized normal CDF, $\varphi_{\text{nor}}(\cdot)$ is the standardized normal probability density function (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:

$$p_L(t) = 1 - \int_{-\infty}^{\infty} \Phi_{\text{nor}}\left[-\frac{g(D_f, t, \beta_1) - \mu_{\beta_2|\beta_1}}{\sigma_{\beta_2|\beta_1}}\right]\frac{1}{\sigma_{\beta_1}}\,\varphi_{\text{nor}}\left(\frac{\beta_1 - \mu_{\beta_1}}{\sigma_{\beta_1}}\right)d\beta_1 \tag{8}$$

2.1.2. Nodes degradation modeling

Degradation may also affect the transmitter nodes in a fiber-optic network [64, 73]. Thus, a light-emitting diode (LED) degradation model is considered for nodes, as LEDs generate light in fiber optic networks [74]. This section considers the modeling for data from accelerated degradation tests on LEDs reported by Pascual et al. [58]. The model detailed in this section is based on Ref. [70]. 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 the time axis only. Standard acceleration models are applied for temperature and current [60, 75, 76]. The Arrhenius transformation is used on junction temperature in degrees Celsius, ° C, (equation (9)) and the Black's law for current acceleration in milliamps, mA, (equation (10)).

$$x_1 = \frac{11605}{T_C + 273.15} \tag{9}$$

$$x_2 = \log(I_{mA}) \tag{10}$$

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 [70]:

$$D_{ijk,N} = 1 + \beta_{1,N}(x_{1k} - x_1^0)\tau_j + \beta_{2,N}(x_{2k} - x_2^0)\tau_j + b_i\tau_j \tag{11}$$

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 [70]. The subscripts "N" are used to refer to the node's degradation path and parameters, to avoid confusion with the degradation path and parameters of links.

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

$$Y_{ijk,N} = D_{ijk,N} + \varepsilon_{ijk,N} = 1 + \beta_{1,N}(x_{1k} - x_1^0)\tau_j + \beta_{2,N}(x_{2k} - x_2^0)\tau_j + b_i\tau_j + \varepsilon_{ijk,N} \tag{12}$$

where $\varepsilon_{ijk,N} \sim N(0, {\sigma_\varepsilon}^2)$ describes the measurement error, with the independence assumption of b_i and across time [70].

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:

$$F_N(t) = \Pr(T \le t) = \Pr\left[D \le D_{f,N}\right] \tag{13}$$

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

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

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

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

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

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

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

Therefore, the nodes' reliability is given as:

$$p_N(t) = 1 - \Phi_{\text{nor}}\left[\frac{D_{f,N} - \mu}{\tau\sigma_b}\right] \tag{14}$$

2.1.3. Bayesian approach for 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:

$$f(\boldsymbol{\theta}|\text{DATA}) = \frac{L(\text{DATA}|\boldsymbol{\theta})f(\boldsymbol{\theta})}{\int L(\text{DATA}|\boldsymbol{\theta})f(\boldsymbol{\theta})d\boldsymbol{\theta}} \tag{15}$$

where the joint prior distribution $f(\boldsymbol{\theta})$ provides the available prior information about the unknown parameters in $\boldsymbol{\theta}$. $f(\boldsymbol{\theta}|\text{DATA})$ is the joint posterior distribution for $\boldsymbol{\theta}$, which combines the information from the data and the prior distribution. $L(\text{DATA}|\boldsymbol{\theta})$ 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 (15) is a normalizing constant that assures that the joint distribution is a proper probability distribution [70].

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 the expert matters, 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 [70]. In this study, Bayesian inference with low-information distributions (which essentially reflects a lack of strong and precisely quantified prior information) will be employed for the initial estimation of parameters.

The computation of the joint posterior distribution for $\boldsymbol{\theta}$ (equation (15)) 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 [70]. A powerful method for simulating a sample from a particular joint posterior distribution is the Markov chain Monte Carlo (MCMC) approach [72]. Gibbs sampling and MCMC [72] 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 [70]. The MCMC simulation proposed is summarized in the following algorithm (2.1). WinBUGS software is an excellent alternative to perform the MCMC simulation [77] and will be used in the case study analysis to estimate the posterior parameters.

Algorithm 2.1:

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 (8) and (14), 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: The available evaluation values of links and nodes reliability provide appropriate data to determine not only point estimates, but also credible intervals. Therefore in this step, from the reliability values evaluated for each time value, obtain the point estimates (or the median values) and the credible intervals (e.g., 95% credible intervals).

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

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:

$$R_{net} = Pr\{(Operational\ links\ connect\ all\ nodes)\ AND\ (all\ nodes\ are\ operational)\} \quad (16)$$

Even considering only link failures and due to the exponential growth of the number of states with the size of networks, the all-terminal reliability calculation is an NP-hard problem [78]. In previous studies based on deep learning techniques, the exact network reliability -calculated with a backtracking algorithm- 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 [14, 48]. For instance, the exact backtracking algorithm reportedly took an average of about 500 seconds per network [48] on ten-node networks.

As an alternative, a MC 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 no 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 an 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.:

$$\hat{R}_{net} = R_{MC} = r/M \quad (17)$$

From equation (16) 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 (16) can be expressed as:

$$R_{net} = Pr\{(Operational\ links\ connect\ all\ nodes)\} \times Pr\{(all\ nodes\ are\ operational)\} \quad (18)$$

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

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:

$$R_{net} = R_{links} \times R_{nodes} \quad (19)$$

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 (20). Consequently, the simulation of nodes is not required.

$$R_{nodes} = p_N^{|N|} \quad (20)$$

On the other hand, regarding the reliability of links (considering the independencies among the nodes and the 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 [12]:

$$R_{links} = \sum_{\Omega} \left[\prod_{j \in L'} p_L \right] \left[\prod_{j \in (L \setminus L')} (1 - p_L) \right] \quad (21)$$

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

Algorithm 2.2:
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 2.2 and equation (20), the network reliability can be estimated as:

$$\hat{R}_{net} = R_{MC_{links}} \times R_{nodes} \tag{22}$$

where $R_{MC_{links}}$ is evaluated by Algorithm 2.2 and R_{nodes} is calculated with equation (20).

2.3. Deep neural network 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 (22), with $R_{MC_{links}}$ evaluated by Algorithm 2.2. The proposed MC algorithm can estimate the network reliability for given (fixed) nodes and links reliability values, even if this algorithm is not currently practical for real-time applications. Moreover, in our approach, there is a need for not only a quick network reliability estimation at each time point, but at each time point, we also consider several (thousands) samples of nodes and links reliability values. Therefore, 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 (20) (to calculate the reliability of nodes R_{nodes}) conform a DNN model. The DNN model is expected to predict the network reliability (for the new given p_L and p_N) accurately and quickly. It is worth to mention that the DNN is trained specifically for a given network (topology). A representation of the DNN model is shown in Figure 2.

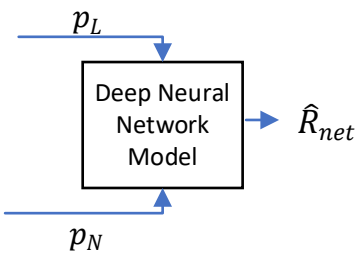


Figure 2. The deep neural network to estimate the reliability of a network

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 the link's reliability values, e.g., {0.01, 0.02, …, 0.99, 1.00}.

2.3.1. DNN architecture

The proposed DNN architecture is based on feed-forward neural networks as they have proven to be effective function approximators [79]. 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 [14, 17]. In addition, a dropout layer is placed before the output layer to avoid overfitting by randomly dropping neurons during the training [80]. To summarize, the architecture is a stack of the following layers: input: (X_t), fully connected hidden layers, dropout, output with sigmoid activation: $(\widehat{y_t})$.

2.3.2. 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.

2.3.3. 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 dependencies are incorporated by the MC Algorithm 2.2 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 (23). Further, the reliability of nodes (R_{nodes}) for a given nodes reliability value (p_N) is given by equation (20). Therefore, the estimated all-terminal reliability of the network is given by the proposed DNN model as:

$$\hat{R}_{net} = \underbrace{DNN(p_L)}_{\hat{R}_{links}} \times R_{nodes}(p_N) = DNN(p_L) \times p_N^{|N|} \tag{23}$$

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

2.4. Bayesian updating of parameters and network reliability prediction

To further reduce the uncertainty in parameter estimates and network reliability prediction obtained from the initial (possibly accelerated degradation test (ADT)) data and initial Bayesian parameter estimation, the framework allows updating the initial estimations with new data. The new data is incorporated according to Algorithm 2.3 proposed below.

Algorithm 2.3:

Step 1: From the initial MCMC draws obtained using Algorithm 2.1, obtain informative prior distributions for parameters of the distributions assumed for random parameters; i.e., informative 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 [70].

Steps 2 – 6: are the same Steps 2-6 as previously defined in Algorithm 2.1.

2.5. Proposed framework for network reliability estimation and updating of parameters

To summarize, the overall proposed framework can estimate the reliability of a network, considering the degradation data of links and nodes. The framework is broadly composed of a links degradation model, a nodes degradation model, and a DNN model (equation (23) and Figure 2) trained using reliability values

obtained by the Monte Carlo Algorithm 2.2. The links degradation model (Figure 3) 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 4. 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.

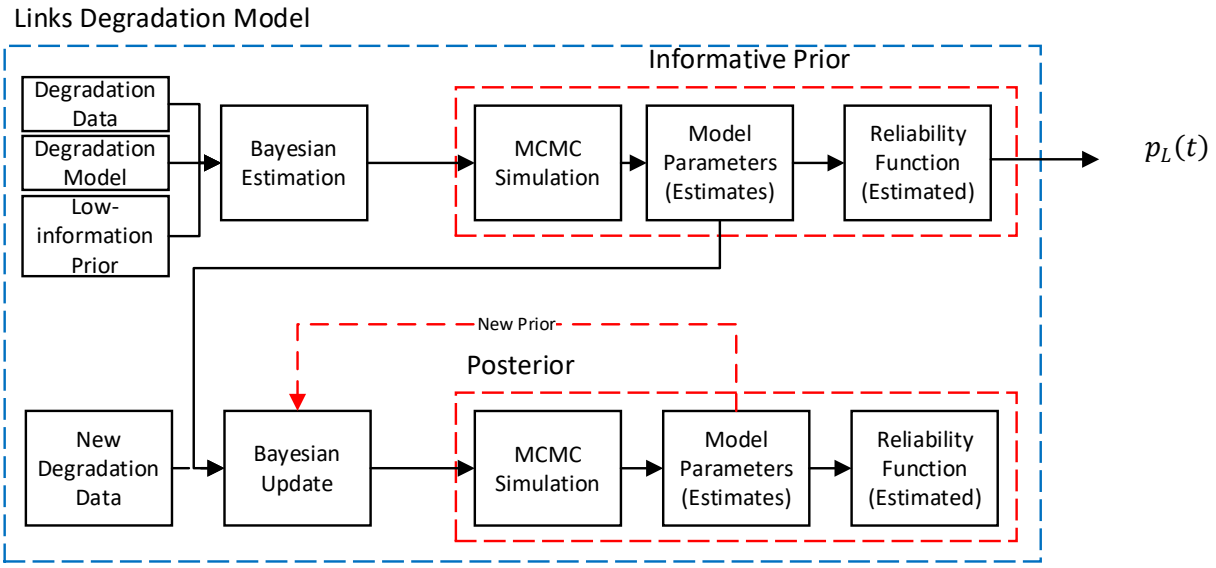


Figure 3. Links degradation model

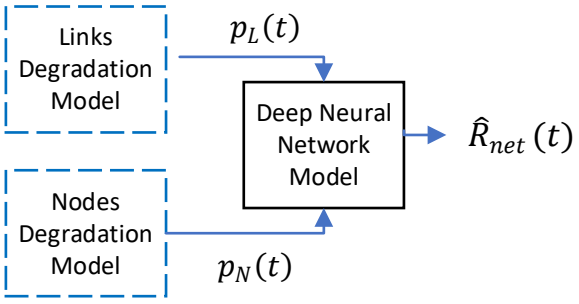


Figure 4. Framework for all-terminal network reliability estimation

3. Case Study

To demonstrate the proposed framework, the real-world network topology Ion (125 nodes, 150 links, New York, USA [56, 57]), shown in Figure 5, was analyzed using the proposed approach (Figure 5). The real degradation data was considered to simulate additional degradation data for links [59]. Similarly, the real degradation data was obtained for nodes from Ref. [58]. 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. [58] 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. Some data were removed to match the data required for 125 nodes, as detailed in Section 3.2.

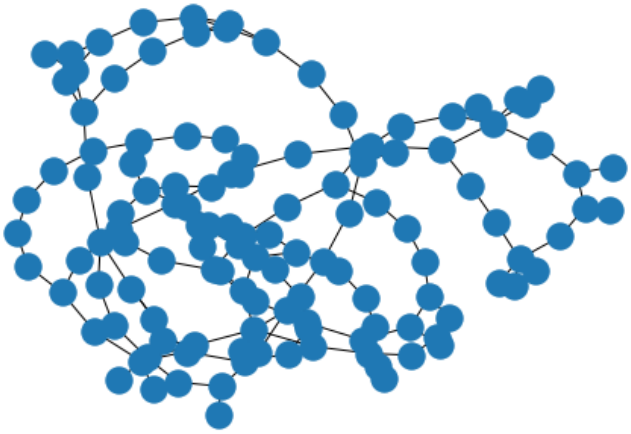


Figure 5. Ion network graphical representation

3.1. 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 6). Considering the degradation model given by equation (6), the random parameters β_1 and β_2 are modeled by a bivariate normal distribution with parameters $\left(\mu_{\beta_1}, \mu_{\beta_2}, \sigma_{\beta_1}, \sigma_{\beta_2}, \rho\right)$,

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.

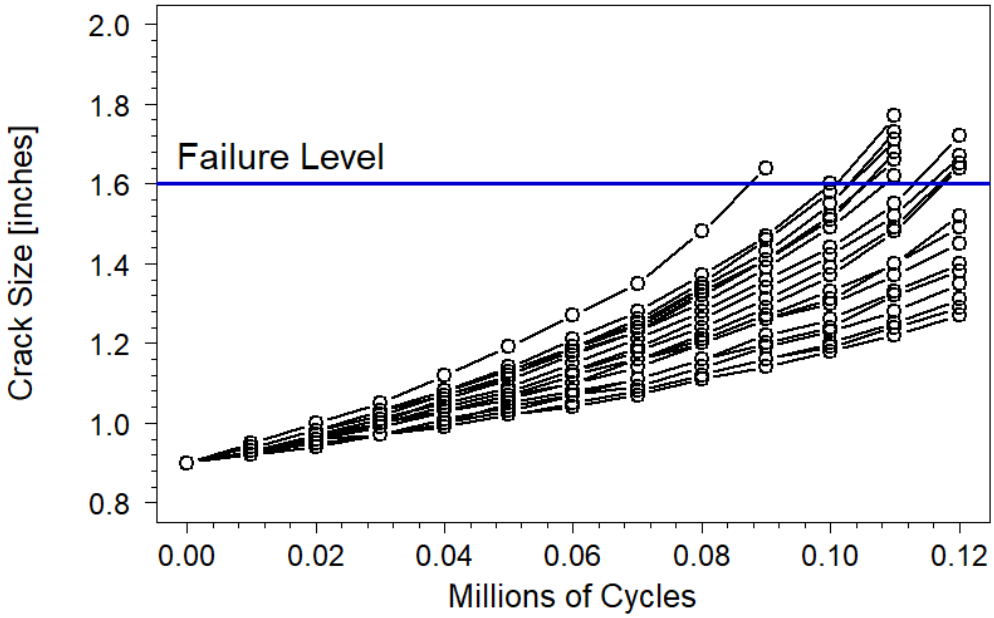


Figure 6. Crack growth data

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:

$$\begin{pmatrix}\beta_1\\ \beta_2\end{pmatrix} \sim N\left(\begin{pmatrix}\mu_{\beta_1}\\ \mu_{\beta_2}\end{pmatrix}, \begin{pmatrix}{\sigma_{\beta_1}}^2 & \rho\sigma_{\beta_1}\sigma_{\beta_2}\\ \rho\sigma_{\beta_1}\sigma_{\beta_2} & {\sigma_{\beta_2}}^2\end{pmatrix}\right) \tag{24}$$

Or equivalently:

$$\begin{pmatrix}\beta_1\\ \beta_2\end{pmatrix} \sim N(\boldsymbol{\mu_\beta}, \boldsymbol{\Sigma_\beta}) \tag{25}$$

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, 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} [81]. In WinBUGS, the multivariate normal distribution is specified in terms of a mean vector and a precision matrix (inverse of the covariance matrix), as shown in equation (26):

$$\boldsymbol{\mu_\beta} = \begin{pmatrix} \mu_{\beta_1} \\ \mu_{\beta_2} \end{pmatrix} \sim N\left(\boldsymbol{\mu_{\mu_\beta}}, \mathbf{T_{\mu_\beta}}^{-1}\right) \tag{26}$$

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

On the other hand, to represent a vague prior knowledge for $\boldsymbol{\Sigma_\beta}$, a low-information Wishart distribution [71, 82] is used to describe the precision matrix $\mathbf{T_\beta}$, where $\mathbf{T_\beta} = \mathbf{\Sigma_\beta}^{-1}$, as shown in equation (27).

$$\mathbf{T_\beta} = \mathbf{\Sigma_\beta}^{-1} \sim W_p\left(\mathbf{S_0}^{-1}, \nu_0\right) \tag{27}$$

The parameters of a Wishart distribution W_p of a $p \times p$ symmetric positive definite matrix is the degrees of freedom ν_0, and the $p \times p$ positive definite scale matrix $\mathbf{S_0}^{-1}$. 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 [71, 81], i.e., $\nu_0 = p$, and $\mathbf{S_0}$ represents a prior guess at the order of magnitude of the covariance matrix $\mathbf{\Sigma_\beta}$ [81].

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 the variance) parameter is used to specify normal distributions. Therefore, using WinBUGS parameters, the residual deviation can be described as:

$$\epsilon_{ij} \sim N(0, {\tau_\epsilon}^{-1}) \tag{28}$$

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:

$$\tau_\epsilon = {\sigma_\epsilon}^{-2} \sim Gamma(\alpha, \beta) \tag{29}$$

A common low-information prior Gamma distribution is obtained by letting $\alpha = \beta = 0.001$ [81], which provides a mean of 1 and a large variance of 1000. Table I summarizes the low-information prior distributions used for the initial estimation of parameters.

TABLE I: LOW-INFORMATION PRIOR DISTRIBUTION FOR LINKS DEGRADATION MODEL

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

To summarize, a WinBUGS model was built considering the original degradation data [59], the degradation model given by equation (6), 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 I). The initial 4,000 MCMC sample draws were dropped ("burn-in") and the sample draws were "thinned" [77] 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 [70]. The point estimates are obtained by the median values from the MCMC sample draws [83]. The results are: $\hat{\boldsymbol{\mu}}_{\boldsymbol{\beta}} = \begin{pmatrix} 3.717 \\ 5.225 \end{pmatrix}$, $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}} = \begin{pmatrix} 0.5219 & -0.1735 \\ -0.1735 & 0.2349 \end{pmatrix}$, and $\hat{\sigma}_\epsilon = 0.008008$. WinBUGS also provides kernel density estimations. As an example, Figure 7 shows the kernel density estimations of parameter $\boldsymbol{\mu_\beta} = \begin{pmatrix} \mu_{\beta_1} \\ \mu_{\beta_2} \end{pmatrix}$.

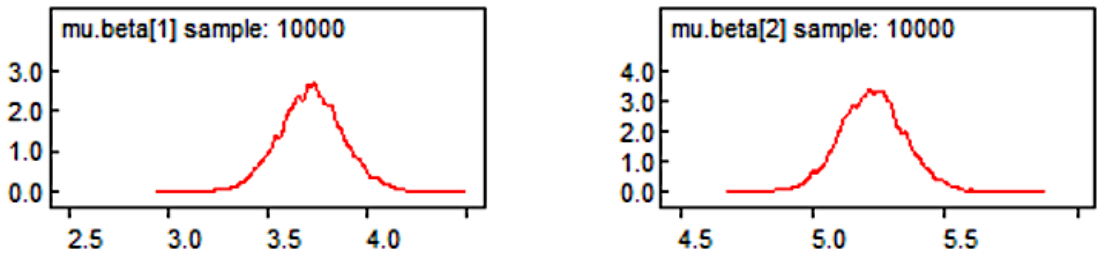


Figure 7. Bayesian kernel density estimation of parameter $\boldsymbol{\mu_\beta}$

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 into 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.

3.1.1. Bayesian estimation of parameters

A WinBUGS model was built considering the "initial data", the degradation model given by equation (6), 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 I. A lag parameter L of 60 was applied. 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 8 shows the kernel density estimations of parameter $\boldsymbol{\mu}_{\boldsymbol{\beta}} = \begin{pmatrix} \mu_{\beta_1} \\ \mu_{\beta_2} \end{pmatrix}$.

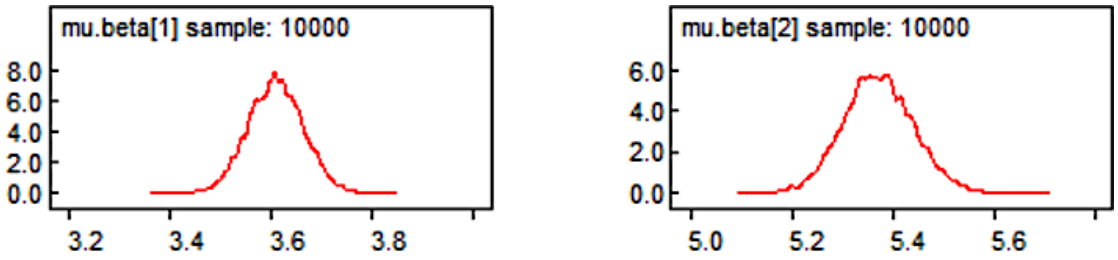


Figure 8. Bayesian kernel density estimation of parameter $\boldsymbol{\mu}_{\boldsymbol{\beta}}$ for "initial data"

3.1.2. Links reliability estimation

Both "initial data" and "new data", for links, have Mcycles as "time" axis, whereas nodes degradation data, considered in Section 3.2 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 (8) to obtain links reliability draws for 1,001 time values between 0 and 10,000 hours. In other words, to plot the link reliability, we considered 1,001 points in the time axis, and for each of those points, there are 10,000 MCMC samples from which the median values and credible bounds were derived and plotted in the reliability axis. The same plotting parameters will be used consistently in the case study to draw nodes reliability and network reliability. The median values (solid line) and the 95% credible bounds (dashed lines) are shown in Figure 9.

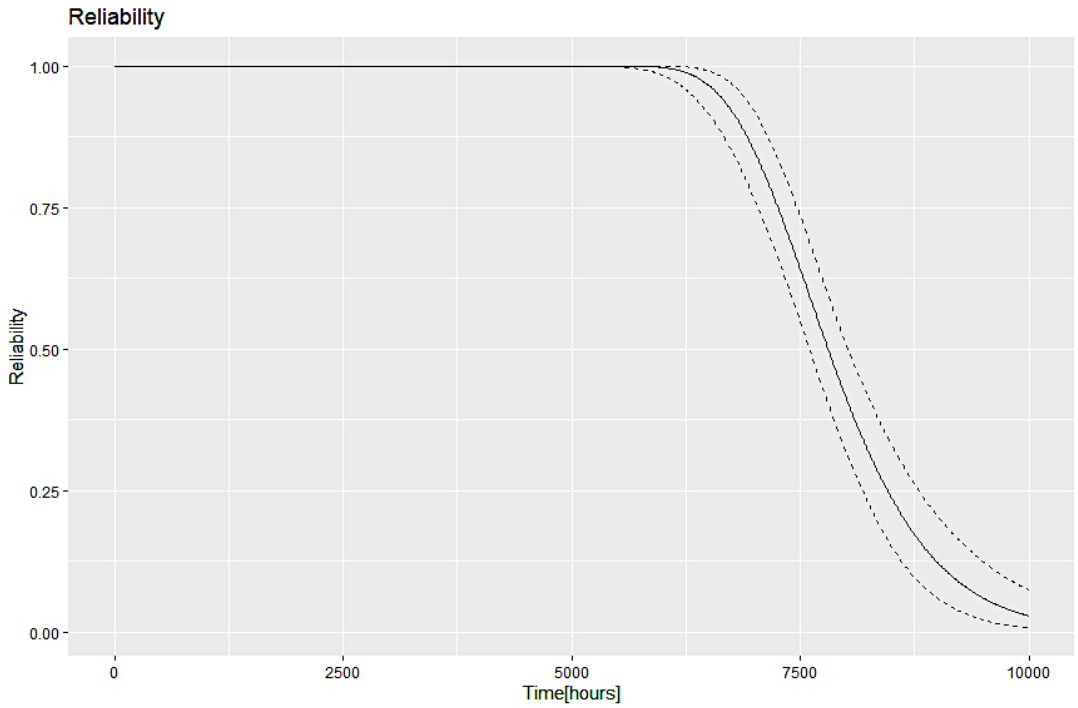


Figure 9. Links reliability and 95% credible intervals

3.1.3. 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 (6), 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 3.1.1 are used to estimate the parameters of informative prior distributions (see Table II). 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 3.1.1). 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 [83] obtained in Section 3.1.1. To make $\boldsymbol{\Sigma_\beta}$ closely centered around $\boldsymbol{\Sigma_0}$, a large ν_0 is selected [71], whereas $\mathbf{S_0}$ is given by equation (30) [71]:

$$\mathbf{S_0} = (\nu_0 - p - 1)\boldsymbol{\Sigma_0} \tag{30}$$

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 lag parameter was L = 100. The point estimates obtained by the median values from the MCMC sample draws are: $\widehat{\boldsymbol{\mu}}_{\boldsymbol{\beta}} = \begin{pmatrix} 3.609 \\ 5.333 \end{pmatrix}$, $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}} = \begin{pmatrix} 0.4354 & -0.1479 \\ -0.1479 & 0.2221 \end{pmatrix}$, and $\hat{\sigma}_\epsilon = 0.008611$. Figure 10 shows the kernel density estimations of parameter $\boldsymbol{\mu_\beta} = \begin{pmatrix} \mu_{\beta_1} \\ \mu_{\beta_2} \end{pmatrix}$.

TABLE II: INFORMATIVE PRIOR DISTRIBUTION SPECIFICATIONS FOR LINKS DEGRADATION MODEL

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

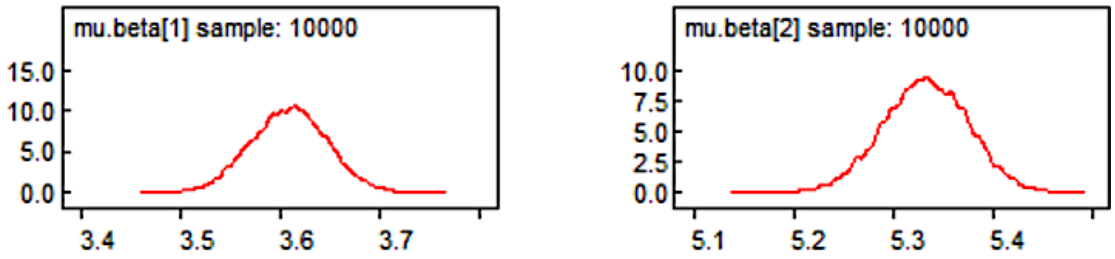


Figure 10. Bayesian kernel density estimation of parameter $\boldsymbol{\mu_\beta}$ for "new data"

To verify the robustness of the posterior distributions, we performed sensitivity analysis by considering a plausible alternative model with changes in informative prior distributions [77]. A natural sensitivity analysis considers longer-tailed alternatives instead of normal distributions [72]. Hence, in the alternative model, a prior distribution for $\boldsymbol{\mu_\beta}$ was specified with the MLE estimates [84] obtained for a multivariate Student's t-distribution from the MCMC draws obtained in Section 3.1.1. In WinBUGS, the (noncentral) multivariate Student's t-distribution is specified in terms of the mean vector, precision matrix (inverse of the 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 III shows the sensitivity of posterior inference in terms of the median and 95% credible intervals obtained from the MCMC sample drawings. Minor differences are observed between the posteriors resulted from the model with normal distribution and the alternative model with the *t* distribution. Therefore, the original model that considers the informative prior distributions in Table II was used for Bayesian updating of parameters.

Table III: Parameters under different prior assumptions

Parameter	Distribution for $\boldsymbol{\mu_\beta}$			
	Multivariate normal		Multivariate Student's t-distribution	
	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	$[0.00798, 0.009315]$

3.1.4. Links reliability estimation updating

The updated 10,000 MCMC draws from the joint posterior distributions of the model parameters were used in equation (8) to compute links reliability draws, considering the same plotting parameters defined in Section 3.1.2. The median values, as well as the 95% credible intervals, are shown in Figure 11. As expected, the informative prior Bayesian updating improved the precision of the estimates. Figure 12 shows both the initial links reliability estimation and credible intervals in solid blue lines. Figure 12 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 improves the credible interval and provides a narrower range compared to the initial credible interval. This is caused by a reduction in uncertainty because of the availability of additional new data.

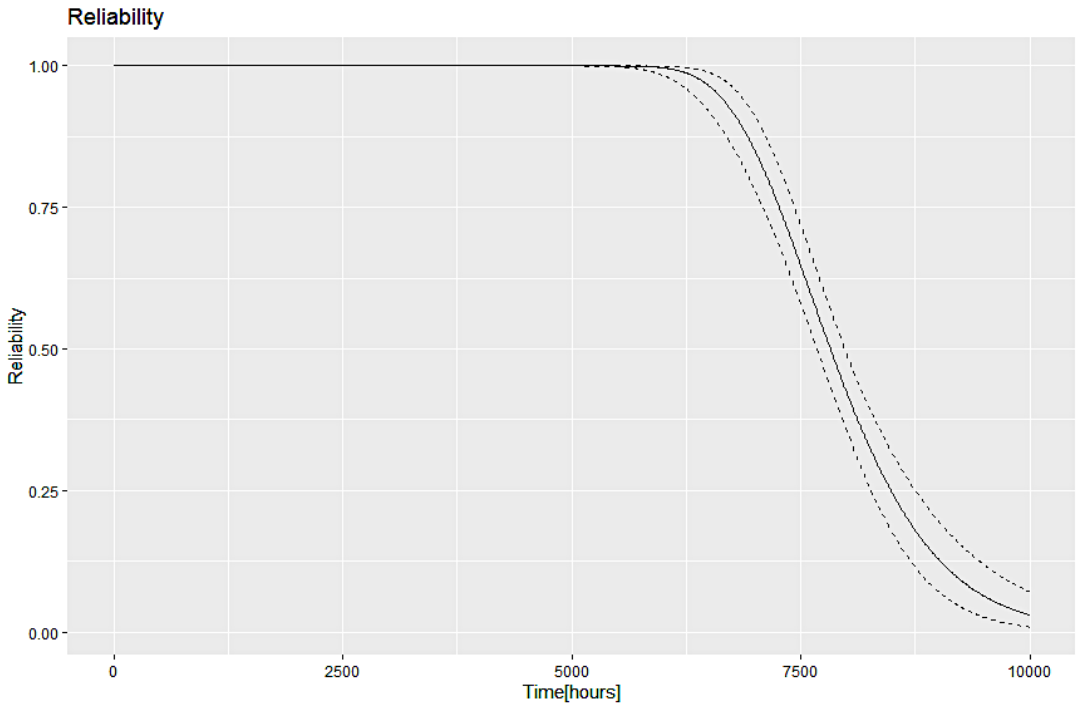


Figure 11. Updated links reliability and 95% credible intervals

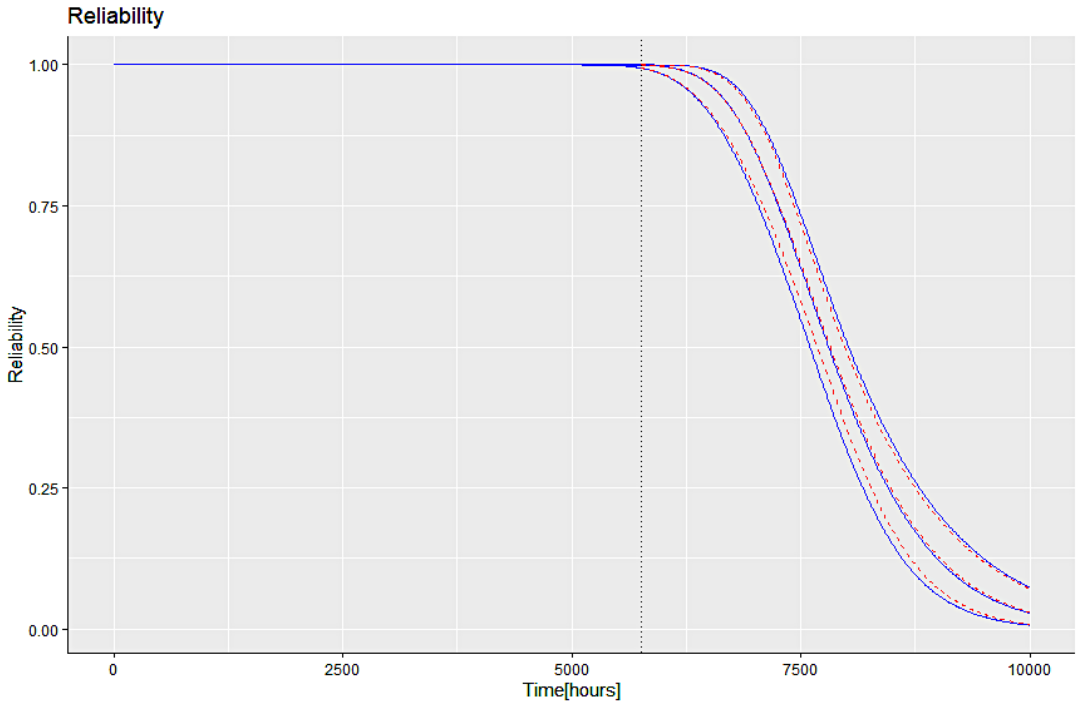


Figure 12. Initial and updated links reliability and 95% credible intervals

3.2. Nodes degradation modeling for reliability evaluation

In this section, normalized (relative to an initial measurement taken on each unit) degradation data [58] for LED light-output are considered, as shown in Figure 13. A decrease in light intensity output with time and LED failure is defined when the relative light intensity output reaches 60% level of the initial value [58]. In the first 138 hours, the sample degradation paths had a complicated irregular behavior for which LED experts had no explanation [58, 70]. Since the 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 14.

The group at 130 °C junction temperature and 40 mA current are believed to cause the occurrence of a new failure mechanism [58]. 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 (3.2.1). Additionally, using the initial estimated parameters, "new data" will be generated to demonstrate the Bayesian parameter updating in Section 3.2.3.

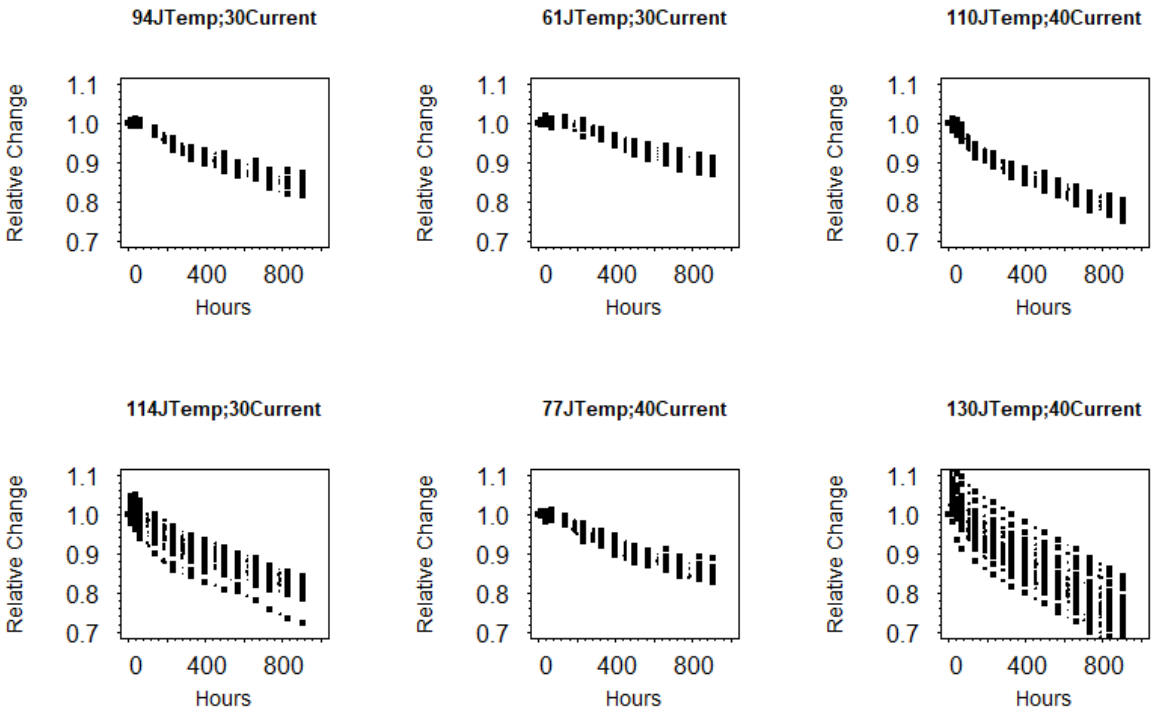


Figure 13. Original normalized LED degradation data

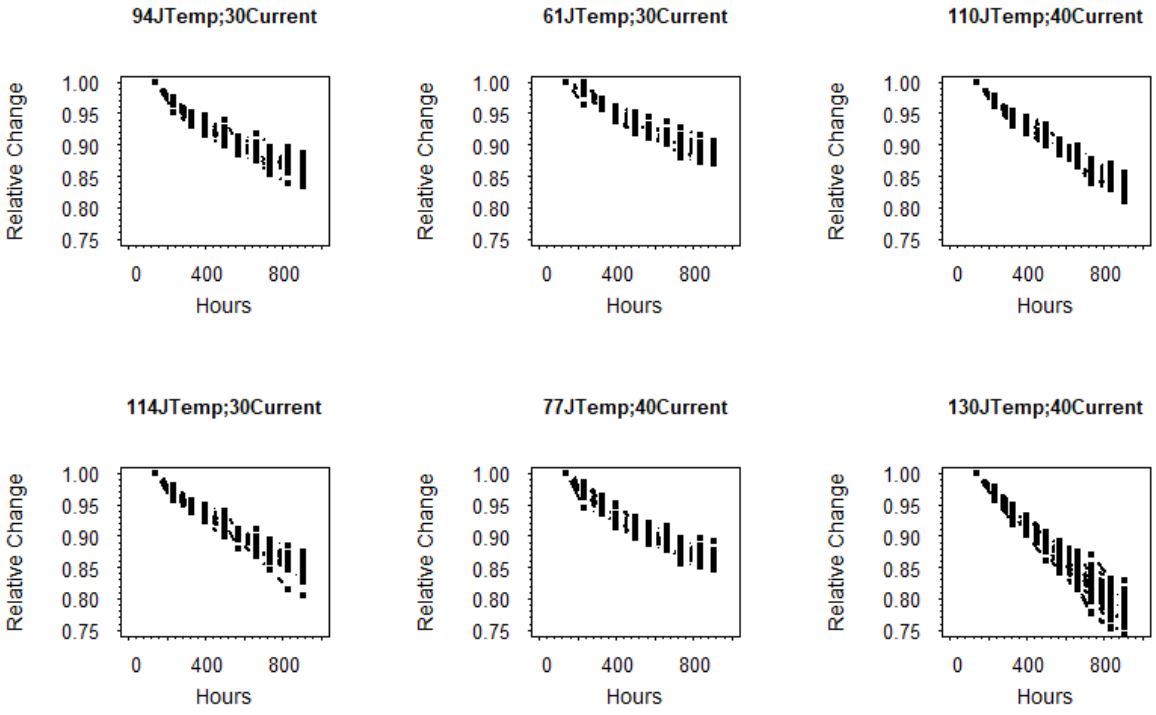


Figure 14. Renormalized LED degradation data

3.2.1. Bayesian estimation of parameters

The degradation model is given by equation (12) 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. Bayesian estimation of parameters was carried out in the same way as for the case of links (Section 3.1.1). Low-information distributions [70] will be considered for such parameters, as shown in Table IV. "Flat" priors correspond to uniform distributions between $-\infty$ and ∞ [70].

TABLE IV: LOW-INFORMATION THE 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 (12), 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 IV. The lag parameter was L of 40. The point estimates obtained by the median values from the MCMC sample draws are: $\hat{\mu}_b = -0.008122$, $\hat{\sigma}_b = 6.04 \times 10^{-4}$, $\beta_{1,N} = 3.961 \times 10^{-4}$, $\beta_{2,N} = -0.002555$, 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 [58]. Figure 15 shows the kernel density estimations of parameter $\hat{\mu}_b$.

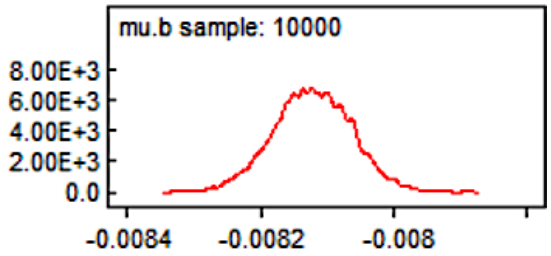


Figure 15. Bayesian kernel density estimation of parameter μ_b for "initial data"

3.2.2. Nodes reliability estimation

The 10,000 MCMC sample draws from the joint posterior distributions of the model parameters were used in equation (14) to obtain nodes reliability curves, considering the same plotting parameters defined in Section 3.1.2. The median values (solid line), as well as the 95% credible bounds (dashed lines), are shown in Figure 16.

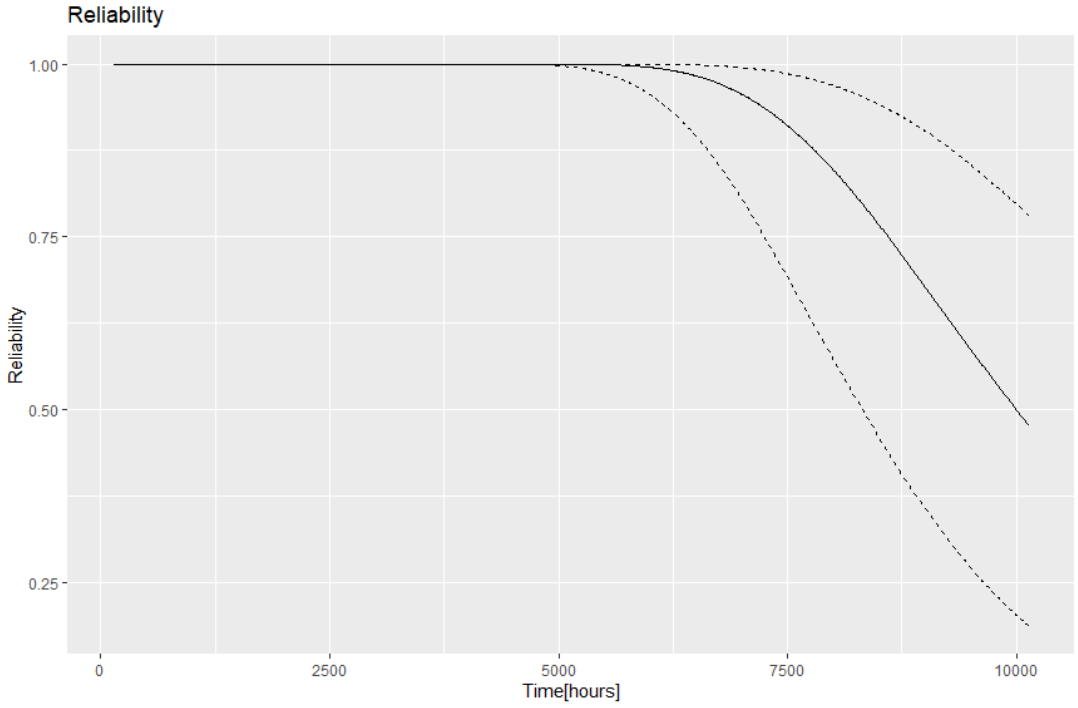


Figure 16. Nodes reliability and 95% credible intervals

3.2.3. 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 (12), the distributions assumed for the parameters b and ε, and informative prior distributions assumed for parameters $\mu_b, \sigma_b, \beta_{1,N}$, and $\beta_{2,N}$. Bayesian estimation was performed in the same way as for the case of links (Section 3.1.3). A lag parameter L of 100

was considered. The point estimates obtained by the median values from the MCMC sample draws are: $\hat{\mu}_b = -0.008117$, $\hat{\sigma}_b = 5.918 \times 10^{-4}$, $\beta_{1,N} = 4.019 \times 10^{-4}$, $\beta_{2,N} = -0.002658$, and $\hat{\sigma}_\epsilon = 0.004668$. Figure 17 shows the kernel density estimations of parameter $\hat{\mu}_b$.

TABLE V: 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	$Uniform(1.0 \times 10^{-5}, 1.0 \times 10^{4})$

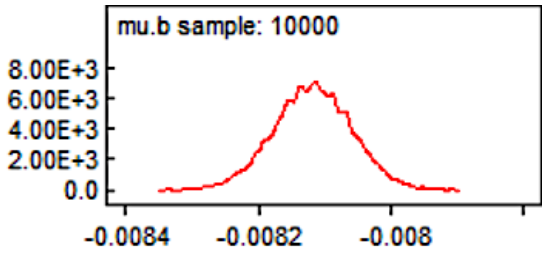


Figure 17. Bayesian kernel density estimation of parameter μ_b for "new data"

Like the nodes' case, we performed a sensitivity analysis to verify the robustness of the posterior distributions. In the alternative model, the prior distributions for $\boldsymbol{\mu_b}$, $\boldsymbol{\beta_{1,N}}$, and $\boldsymbol{\beta_{2,N}}$ are given as $\boldsymbol{t(-0.008121633, 301507501^{-1}, 100)}$, $\boldsymbol{t(0.000396097, 1059545212^{-1}, 46)}$, and $\boldsymbol{t(-0.002557149, 6460993^{-1}, 100)}$, respectively. Table VI shows the sensitivity analysis results. Due to minor differences between the results obtained with original and alternative models, the original model that considers the informative prior distributions in Table V was used for Bayesian updating of parameters.

TABLE VI: POSTERIOR MEDIAN AND 95% CREDIBLE INTERVALS OF PARAMETERS UNDER DIFFERENT PRIOR ASSUMPTIONS

Parameter	Distributions for μ_b, $\beta_{1,N}$, and $\beta_{2,N}$	
	Normal	Student's t- Distribution

	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	$[0.004350, 0.005034]$

3.2.4. Nodes reliability estimation updating

The updated 10,000 MCMC draws from the joint posterior distributions of the model parameters were used in equation (14) to compute the nodes reliability curves, considering the same plotting parameters defined in Section 3.1.2. Similar to what occurred in the case of links reliability, figures 18 and 19 show narrower credible intervals after updating.

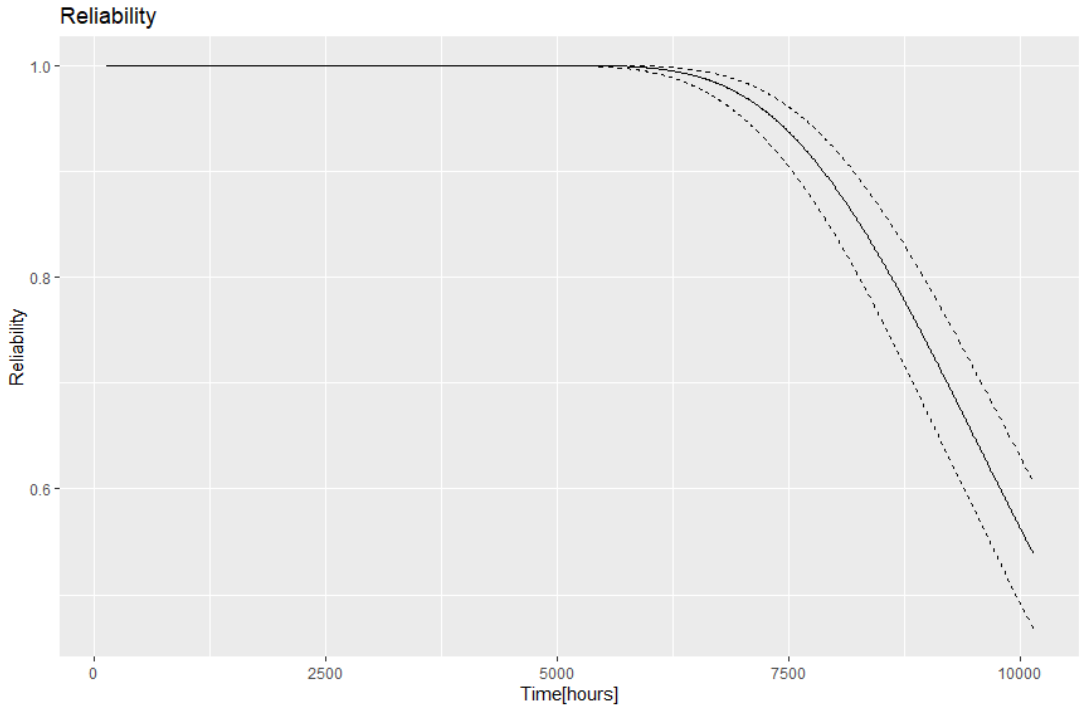


Figure 18. Updated nodes reliability and 95% credible intervals

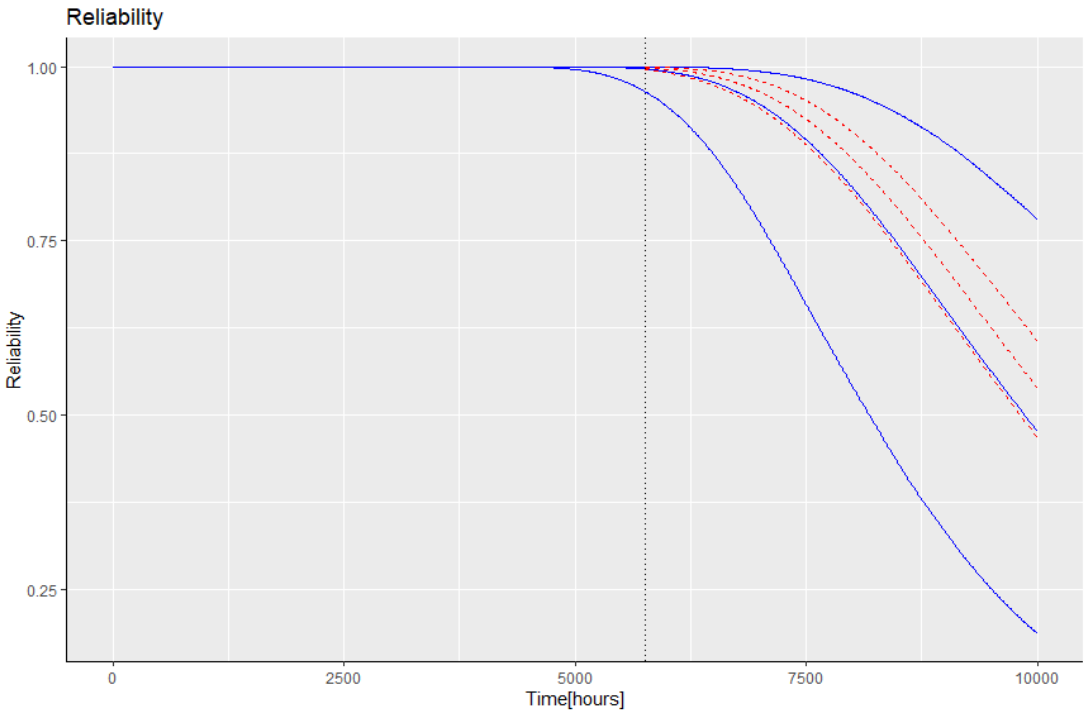


Figure 19. Initial and updated nodes reliability and 95% credible intervals

3.3. MC method and DNN model for all-terminal network reliability estimation

In this section, the use of the MC method and DNN to evaluate the network reliability is illustrated. MC method (Algorithm 2.2) 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 (23) for any given p_L and p_N.

A dataset of 100 link reliability values is considered, i.e., {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. 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 into 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 [85]. 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 [86] 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 VII, column 2). The average cross validation-error is given by equation (31) [14, 48].

$$RMSE_{cv=}\sqrt{\frac{1}{100}\sum_{g=1}^{5}\sum_{h=1}^{20}\left(y_{(g-1)\times 20+h}-\hat{y}_{(g-1)\times 20+h}\right)^2} \tag{31}$$

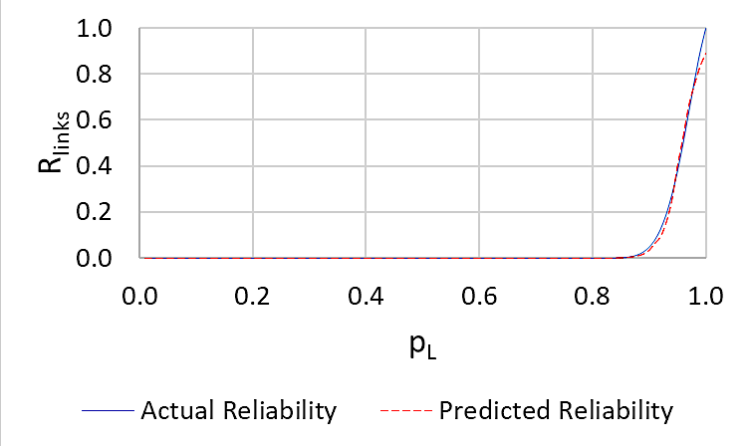


Figure 20. Predicted ($\widehat{R}_{links}$) and actual ($R_{MC_{links}}$) reliability of links, as a function of links reliability (p_L)

TABLE VII: 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 RMSE measures the accuracy of the DNN prediction. The error (0.01460) outperforms previous results achieved by ANN-based approaches, e.g., RMSE of 0.06260 [14] and RMSE of 0.04406 [48]. 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 VII, columns 3, 4) for the mean difference show no significant pairwise difference between actual and the predicted values. Therefore, the DNN provides a good fit, as shown in Figure 20. Figure 20 also shows that the predicted values (R_{links}) noticeably underestimate the actual reliability ($R_{MC_{links}}$ calculated with Algorithm 2.2) 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 (23) when applied for network reliability estimation.

The hierarchical approach allows a smooth fit even at high-reliability values, as shown in Figure 21. Figure 21 provides a graphical view of the performance of the DNN model to predict the network reliability ($\hat{R}_{net}$) as a function of the links and nodes reliability values, i.e., p_L and p_N, respectively. The estimated network reliability for a combination of 10,000 values for both p_L and p_N uniformly distributed between 0 and 1 is plotted in Figure 21. As we can expect, both p_L and p_N, have effect in $\hat{R}_{net}$. Figure 21 shows that 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. In contrast, a link failure may be alleviated by communication through other surviving links. Therefore, the network reliability $\hat{R}_{net}$ is more sensitive to the nodes reliability p_N than to the links reliability p_L

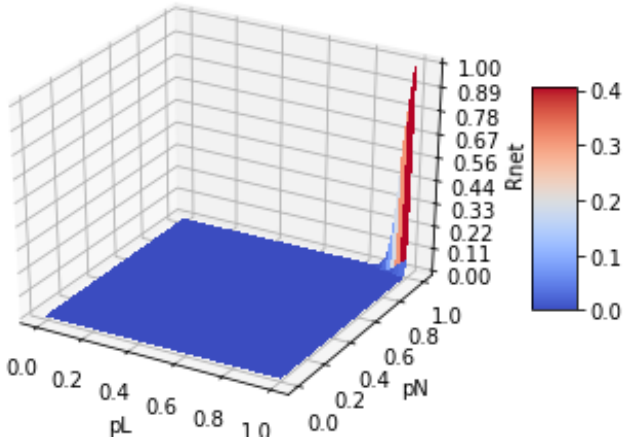


Figure 21. Estimated reliability of networks ($\widehat{R}_{net}$), as a function of links reliability (p_L) and nodes reliability (p_N)

There is a significant computation time difference between the pure MC algorithm and the integrated framework based on MC-DNN (Table VII, columns 5, 6). Table VII (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., $\frac{3{,}162s}{10{,}000 \times 1{,}001}$). It is worth mentioning that the execution time (0.316ms) does not consider the training time, which is in the order of 60s, but it has to be done only once for a given network. 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.

3.4. Network reliability estimation

In this section, the network reliability estimation is illustrated. The links reliability draws (Section 3.1.2), and the nodes reliability draws (Section 3.2.2) were fed to the hierarchical DNN model (equation (23) with the appropriate DNN) to obtain network reliability curves, considering the same plotting parameters defined in Section 3.1.2. The median values (solid line), as well as the 95% credible bounds (dashed lines), are shown in Figure 22.

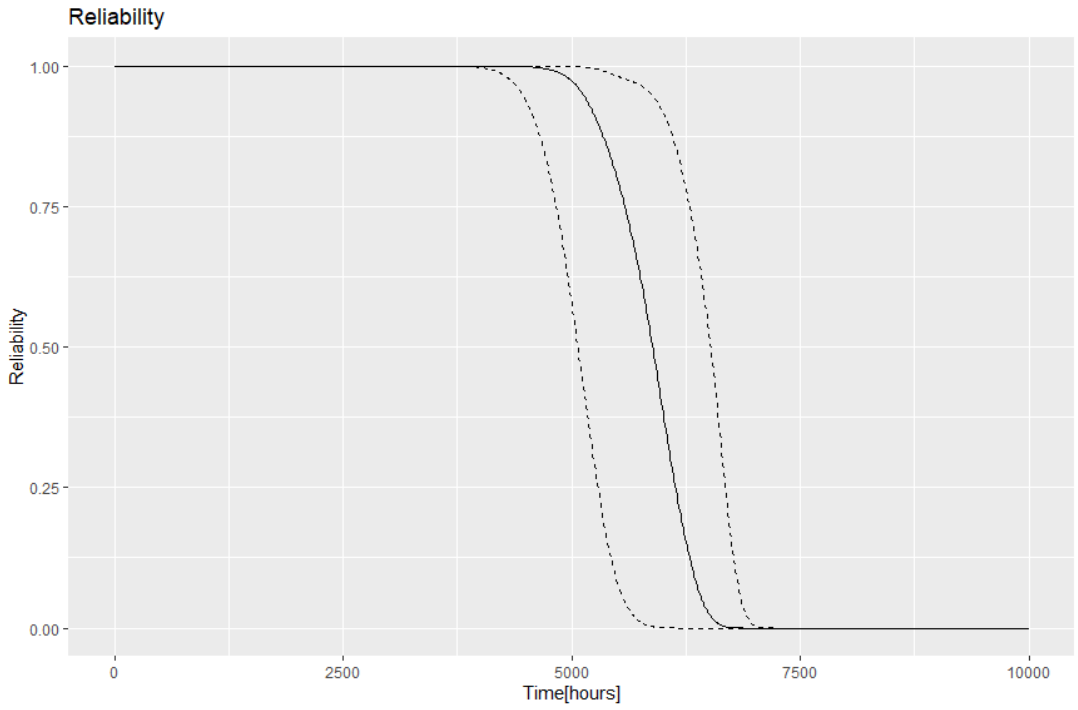


Figure 22. Network reliability and 95% credible intervals

The sharp decline of network reliability between 5,000 and 7,500 hours approximately (Figure 22) could be explained because a rapid decrease of links reliability starts around 6,000 hours (Figure 12), and a similar situation occurs with nodes reliability (Figure 19). Moreover, this behavior is expected due to the high sensitivity of network reliability to nodes reliability $\boldsymbol{p_N}$, shown in Figure 21 and explained by equations (20) and (22).

3.5. Network reliability estimation updating

Once additional information becomes available, the updated links reliability draws (discussed in Section 3.1.4) and nodes reliability draws (discussed in Section 3.2.4) 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 23. As expected, the informative prior Bayesian updating improved the precision of the estimates. Figure 24 shows the initial network reliability estimation with credible intervals in solid blue lines. Figure 24 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.

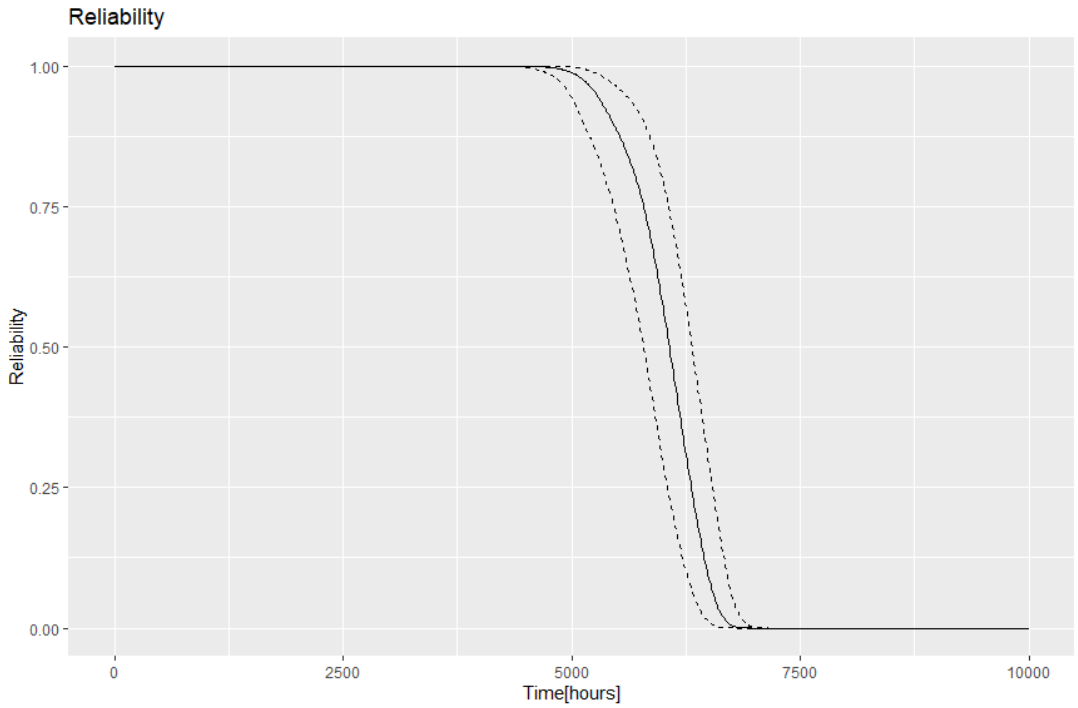


Figure 23. Updated network reliability and 95% credible intervals

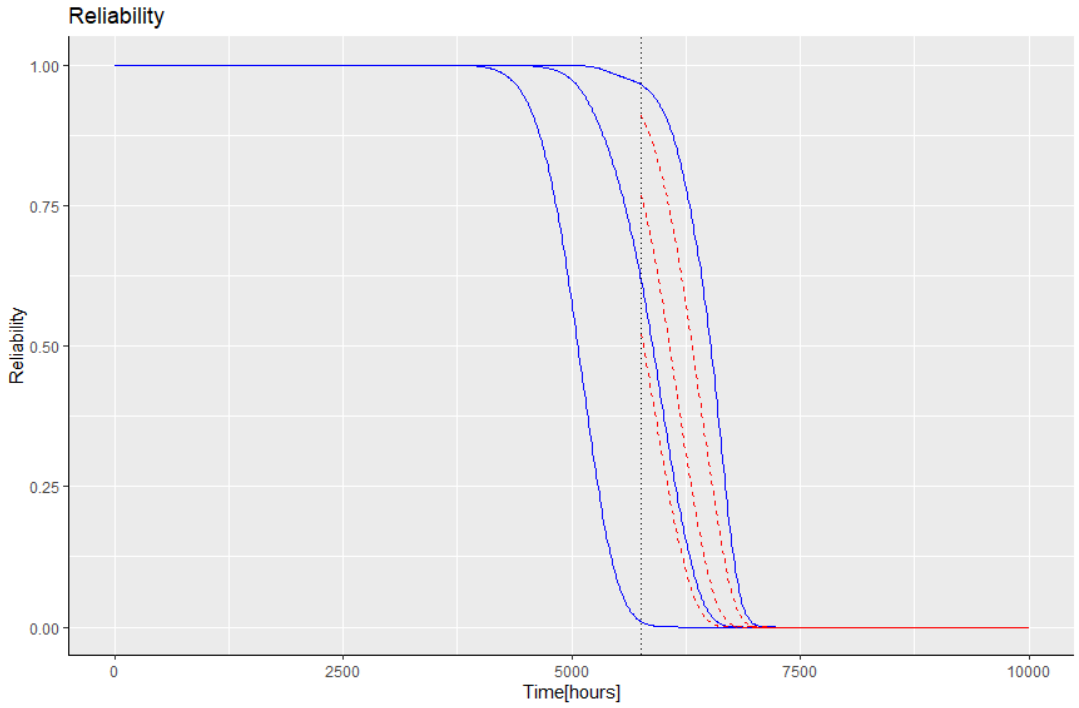


Figure 24. Initial and updated network reliability and 95% credible intervals

4. 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 paper has proposed a framework that accounts for the dynamic behavior of a network by using the degradation data from both links and nodes of the network 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 estimations 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, the integration of MC-DNN with the Bayesian approach provides an 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 estimations and updating are required, such as an online reliability monitoring system. Based on (usually limited) initial accelerated degradation test data, the framework could provide reliability estimates as a function of time. Furthermore, suppose during the normal operation, links, nodes, or both change their degradation profile. In that case, 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. A future research direction may include remaining useful life (RUL) estimation and identifying critical elements of a network.

Acknowledgment
This research is partially supported by Grant No. FAR0025840 by Uniqarta, Inc. (NSF SBIR) and NDSU Center for Quality, Reliability, and Maintainability Engineering (supported by member companies Bobcat, Horton, Inc., and John De-ere) at North Dakota State University.

Funding Data
Center for Quality, Reliability, and Maintainability Engineering (CQRME) at North Dakota State University (Funder ID: 10.13039/100006351).

References:

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

[2] I. Gertsbakh and Y. Shpungin, *Network reliability and resilience*. Springer Science & Business Media, 2011.

[3] 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.

[4] D. Nykamp. "An introduction to networks." https://mathinsight.org/network_introduction (accessed Aug 6, 2019).

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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

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

[12] E. Zio, "Solving advanced network reliability problems by means of cellular automata and Monte Carlo sampling," *Reliability Engineering & System Safety,* vol. 89, no. 2, pp. 219-226, 2005.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] P. Bellavista, *TELECOMMUNICATION SYSTEMS AND TECHNOLOGIES-Volume II*. EOLSS Publications, 2009.

[19] H. Cancela, G. Rubino, and M. E. Urquhart, "An algorithm to compute the all-terminal reliability measure," *OpSearch,* vol. 38, no. 6, pp. 567-579, 2001.

[20] J.-H. Park, "All-terminal reliability analysis of wireless networks of redundant radio modules," *IEEE Internet of Things Journal,* vol. 3, no. 2, pp. 219-230, 2015.

[21] A. Peiravi and H. T. Kheibari, "Fast estimation of network reliability using modified Manhattan distance in mobile wireless networks," *Journal of Applied Sciences,* vol. 8, no. 23, pp. 4303-4311, 2008.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

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

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

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

[32] D. Li, Q. Zhang, E. Zio, S. Havlin, and R. Kang, "Network reliability analysis based on percolation theory," *Reliability Engineering & System Safety,* vol. 142, pp. 556-562, 2015.

[33] W.-C. Yeh, "Novel Algorithm for Computing All-Pairs Homogeneity-Arc Binary-State Undirected Network Reliability," *arXiv preprint arXiv:2105.01500,* 2021.

[34] A. Alkaff, M. N. Qomarudin, and Y. Bilfaqih, "Network reliability analysis: matrix-exponential approach," *Reliability Engineering & System Safety,* vol. 212, p. 107591, 2021.

[35] P.-C. Chang, D.-H. Huang, Y.-K. Lin, and T.-P. Nguyen, "Reliability and maintenance models for a time-related multi-state flow network via d-MC approach," *Reliability Engineering & System Safety,* vol. 216, p. 107962, 2021.

[36] W.-C. Yeh, "A quick BAT for evaluating the reliability of binary-state networks," *Reliability Engineering & System Safety,* vol. 216, p. 107917, 2021.

[37] W.-C. Yeh, "Novel binary-addition tree algorithm (BAT) for binary-state network reliability problem," *Reliability Engineering & System Safety,* vol. 208, p. 107448, 2021.

[38] 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.

[39] R. Jan, "«Design of reliable networks), Computers and Operations Research. Vol. 20," 1993.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

[51] 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.

[52] J. Kawahara, K. Sonoda, T. Inoue, and S. Kasahara, "Efficient construction of binary decision diagrams for network reliability with imperfect vertices," *Reliability Engineering & System Safety,* vol. 188, pp. 142-154, 2019.

[53] S.-Y. Kuo, F.-M. Yeh, and H.-Y. Lin, "Efficient and exact reliability evaluation for networks with imperfect vertices," *IEEE Transactions on Reliability,* vol. 56, no. 2, pp. 288-300, 2007.

[54] A. Satyanarayana and M. K. Chang, "Network reliability and the factoring theorem," *Networks,* vol. 13, no. 1, pp. 107-120, 1983.

[55] A. Satyanarayana and R. K. Wood, "A linear-time algorithm for computing k-terminal reliability in series-parallel networks," *SIAM Journal on Computing,* vol. 14, no. 4, pp. 818-832, 1985.

[56] 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.

[57] S. Knight. "The Internet Topology Zoo." The University of Adelaide. http://www.topology-zoo.org (accessed 3/16, 2021).

[58] 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.

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

[61] 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.

[62] J. Pinheiro, D. Bates, S. DebRoy, D. Sarkar, S. Heisterkamp, and B. Van Willigen, "Package 'nlme'. Linear and Nonlinear Mixed Effects Models," 2019.

[63] ITU, *Optical Fibres, cables and systems*. 2010.

[64] 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.

[65] 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.
[66] 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.
[67] S. Flint, "Failure rates for fiber optic assemblies," IIT RESEARCH INST CHICAGO IL, 1980.
[68] N. E. Dowling, *Mechanical behavior of materials: engineering methods for deformation, fracture, and fatigue*. Pearson, 2012.
[69] D. Steadman, R. Carlson, and G. Kardomateas, "On the form of fatigue crack growth formulae," *International journal of fracture,* vol. 73, no. 4, pp. R79-R81, 1995.
[70] 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.
[71] P. D. Hoff, *A first course in Bayesian statistical methods*. Springer, 2009.
[72] A. Gelman, J. B. Carlin, and H. S. Stern, *Bayesian Data Analysis*, 3rd ed. (Texts in Statistical Science). Boca Raton: CRC Press, 2013.
[73] 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.
[74] G. P. Agrawal, *Fiber-optic communication systems*. John Wiley & Sons, 2012.
[75] W. Nelson, *Accelerated Testing Statistical Models, Test Plans, and Data Analysis*. New York: Wiley, 2004.
[76] L. Escobar and W. Meeker, "A Review of Accelerated Test Models," *Statistical Science,* vol. 21, pp. 552-577, 2006.
[77] I. Ntzoufras, *Bayesian modeling using WinBUGS*. John Wiley & Sons, 2011.
[78] M. R. Garey, "David S. Johnson Computers and Intractability," *AGuide to the Theory of NP-Completeness WH Freeman and CompanyNew York,* 1979.
[79] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
[80] 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.
[81] 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.
[82] 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.
[83] R. Billinton and W. Li, *Reliability assessment of electric power systems using Monte Carlo methods*. Springer Science & Business Media, 2013.
[84] 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.
[85] J. Brownlee, *Better Deep Learning: Train Faster, Reduce Overfitting, and Make Better Predictions*. Machine Learning Mastery, 2018.
[86] 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.