International Journal of Mathematical, Engineering and Management Sciences
Vol. 11, No. 2, 1050-1073, 2026
https://doi.org/10.33889/IJMEMS.2026.11.2.043

ANN- Assisted Reliability and Availability of a Warm Standby Repairable Computer Network System

P. K. Poonia
Mathematics Section, University of Technology and Applied Sciences- Ibri, Ibri-516,
Al Dhahirah Governorate, Sultanate of Oman.
Corresponding author: praveen.poonia@utas.edu.om

Carlton Azeez
Mathematics Section, University of Technology and Applied Sciences- Ibri, Ibri-516,
Al Dhahirah Governorate, Sultanate of Oman.
E-mail: carlton.aziz@utas.edu.om

Suresh Rasappan
Mathematics Section, University of Technology and Applied Sciences- Ibri, Ibri-516,
Al Dhahirah Governorate, Sultanate of Oman.
E-mail: suresh.rasappan@utas.edu.om

(Received on November 23, 2025; Revised on January 13, 2026 & February 12, 2026; Accepted on February 13, 2026)

Abstract

This study investigated the reliability and availability of a computer lab network consisting of three computer labs connected through a server in a parallel configuration under a 2-out-of-3: G policy utilizing a random process, employing a supplementary variable technique, and an artificial neural network (ANN) approach. The proposed complex system may experience failure because of the failure of at least two computer labs, a server, or a catastrophic event occurring at any given time t. The failure rates of the units were constant and expected to follow an exponential distribution. The repair rates are assumed to be general; however, a completely failed system is coupled using the Gumbel-Hougaard family copula. Each type of failure and repair rate is regarded as a neural weight when analyzed using the ANN approach, which is managed using an exponential distribution. The system state probabilities, up and down state probabilities, and availability and reliability of the model are evaluated using a Markovian process and Laplace transforms. Finally, the reliability and availability of the model were analyzed using an ANN framework to estimate the analytical curves. The results obtained using MAPLE software (analytical computation) and MATLAB software (ANN training and simulation) confirm that the ANN provides reliable estimates of the reliability and availability measures that closely match the analytical outcomes.

Keywords- k-out-of-n: G system, Availability, Neural network, Neural weights, Gumbel-Hougaard family copula.

1. Introduction

Computer laboratories play a vital role in scientific learning by providing students with hands-on knowledge and practical skills for their future careers. These labs function as an interactive learning environment for theoretical understanding and real-world use, and hence connect the gap between lecture theater instructions and industry needs. At this juncture, students gain competence in many computer software and programming languages that are essential to computer science and information technology (Singh et al., 2021). Computer labs provide authorization for resources that may not be freely accessible to students outside the educational establishment. It involves dedicated software, secure networks, and high-performance computers. As students frequently collaborate in group projects and exchange ideas and methods, collaborative ambience in computer laboratories also promotes teamwork, peer learning, research, and innovation (Saadon et al., 2011). Recent advancements in information and communication technology are rooted in the development of computer laboratory networks. Hence, computer laboratory networks are critical suppliers for educational and industrial institutions seeking to equip learners with these materials. In a computer laboratory network, hubs, switches, routers, wireless access points, firewalls, cables, patch panels, and terminal nodes are used for PCs and laptops. Therefore, network dependence is crucial. Connectivity reliability is a primary research topic in the field of network reliability. The concept of network performance reliability has been advocated, and network performance has received increasing attention (Poonia, 2021, 2022). Many authors have studied the various reliability features of different computer lab networks and have proposed methods to improve their reliability and availability. In the past, many studies on the reliability and availability of computer lab networks have been published, resulting in substantial literature. Several researchers have published papers including availability analysis of two k-servers under k-out-of-n: F scheme (Tamegai, 1980), reliability optimization using the heuristic approach of a computer communication network (Kiu and McAllister, 1988), all terminal network reliability using recursive truncation algorithm and a bounding approximation algorithm (Sharafat and Ma'rouzi, 2009), marginal reliability importance for network systems with k-terminal reliability (Koide et al., 2005), reliability concepts on various types of maintenance (Nakagawa, 2005), general network reliability that follows two terminal reasoning using the substitution algorithm (Gebre and Ramirez-Marquez, 2007), network reliability for computer networks with node failure under cost/budget constraints (Lin, 2007, 2012), Optimizing maintenance strategies and achieving cost effectiveness of a standby system integrated with cost and sensitivity analysis (Ram et al., 2025b), and k-terminal network reliability using recursive truncation algorithm (Yeh et al., 2002), highlights energy constraints, node failures and scalability together with cross layer reliability strategies and energy efficient fault recovery (Mahmood et al., 2015) and many others. Recently, Zhang et al. (2014) focused on a method that uses graph transformation to lower the complexity of reliability evaluation by scaling down the networks. They considered the design of reliable protocols in their study from 2015 onwards. The authors examined different approaches to maintaining reliability by retransmission or extra copies and using combinations of packet or event reliability to recover lost data using either a hop-by-hop or an end-to-end process. Ding et al. (2016) introduced a new method for improving the reliability of online networks. Sharma et al. (2025) showed a repairable k-out-of-n system using a weighted exponential Lindley distribution and modeled component life-times. This study reveals the applicability of the weighted exponential Lindley distribution through examples and highlights its accuracy over traditional models.

In previous studies, the authors researched various types of networking models focusing on reliability, including the concept of 'repair', encompassing all types of fixes, which is a bounded view of repair. The overall approach adopted for repair was neither optimistic nor economical because the system was required to stop functioning during repair. Even with extensive computer lab network reliability modeling, analysis, and other advanced research domains, there is an evident lack of effort in integrating copula frameworks of performance prediction, assessment, and optimization for the comprehensive and reliable monitoring of computer networks. Realistically, however, one can only hope to achieve online or offline repairs while in a partial failure state without disruption, and if in full failure, an enhanced speed of completion is required through the Gumber-Hougaard family copula (Nelsen, 2006). This belongs to the Archimedean family copula known for its dependence structure, which makes it suitable for this purpose. It is distinct because it is defined by its generator function, which differentiates it from the other copulas. The exponential Gumbel-Hougaard family copula is expressed as:

$$\mu_0\left(x\right)=\exp\left[x^{\theta}+\left\{\log\phi\left(x\right)\right\}^{\theta}\right]^{1/\theta} \text{ where, } u_1\left(x\right)=\phi\left(x\right) \text{ and } u_2\left(x\right)=e^{x}.$$

where, $1<\theta<\infty$ is the strength of the dependence between the variables. Numerous researchers have used the copula distribution to repair completely failed systems and have obtained encouraging results. Ram et al. (2013) were among the first to explore how the Gumber-Hougaard family copula can be used to compute a completely failed distribution. The authors compared two systems that switch devices and used SVT and Laplace transforms to analyze all reliability metrics. Munjal and Singh (2014) analyzed two independent systems that follow the k-out-of-n rule for various r and k values and determined the system's behavior during the steady state. Therefore, we can study how different combinations of repairable and non-reparable, identical and non-identical components influence demand. Chand and Singh (2026), Raghav et al. (2020) and Poonia and Tyagi (2026) explored the performance of a redundant k-out-of-n system in various cases and several failures using the Gumbel-Hougaard family copula. The authors proved that copula repair is preferable because it yields better reliability, availability, and cost investigation than general repair. Nehra et al. (2023) and Poonia (2024) examined various aspects of reliability for a working model in precision agriculture using the supplementary variable technique and the Gumbel-Hougaard family copula repair. The authors formulated a reliability formula and established that copula repair surpasses general repairs. Kumar et al. (2024) modeled an automated waste sorting robotic arm system, and Jadhav and Kumar (2025) exhibited a feeding system using a Markov decision process and enhanced reliability, availability, and profitability. Furthermore, using sensitivity analysis, the authors determined which component failure significantly impacts the overall system performance, and which component reliability improvements grant the maximum gains in the performance of the system. Ram et al. (2025a) suggested a reliability and sensitivity model for a maintainability energy system by applying priority repair policies employing a Markov process and a supplementary variable technique. Key parameters such as reliability, availability, and cost-effectiveness are evaluated, and simultaneously, a test is conducted for sensitivity to failure and repair rates that considerably influence the system performance.

Furthermore, the previously highlighted authors did not employ a neural network approach to investigate the system utilizing the copula distribution and supplementary variable technique. The construction of artificial neural networks is based on biological neural systems, of which the neural network is one of the most important parts of the human brain. This class of ANN is distinguished by self-learning, self-updating, and self-adapting features. It is necessary to train an ANN, following which the network can be employed to solve previously unknown problems. An ANN is composed of a set of processing elements arranged in layers, which include an input layer, one or more hidden layers, and an output layer (Chantola et al., 2020). AI is used worldwide for pattern recognition. Only a handful of authors have addressed the use of reliability models based on ANN modeling. According to Karunanithi et al. (1992), neural network reliability growth models have an advantage over analytic models because they need only failure data as input, unlike the latter, which requires numerous pre-existing assumptions. Furthermore, the authors noted that with the use of failure data, the ANN developed its own failure model and based on this model, it predicted future failures. Gupta et al. (2022) focused on applying an ANN to real-life systems and refined several aspects of reliability. Chau (2007) presented the use of an ANN in the design of structures and formulated an implicit limit state function for reliability assessment.

The remainder of this paper is organized as follows. Section 2 presents system descriptions, assumptions, and notations. Section 3 formulates the model under the adopted operating policy and derives the governing equations. Section 4 describes the boundary/initial conditions and the analytical solution steps. Section 5 introduces the ANN framework used to estimate analytical reliability and availability curves. Section 6 reports the numerical results and comparisons, and Section 7 concludes the work.

2. Model Description and Notations

2.1 System Description

It is clear from the introduction that ANN modeling plays a pivotal role in predicting the future failure of various types of warm redundant complex engineering systems. Moreover, many scholars have investigated k-out-of-n: F/G system models using various methods, including the Gumbel-Hougaard family copula methodology, but very few authors have studied the Gumbel-Hougaard family copula distribution together with ANN modeling. In this respect, we designed a computer lab network consisting of three computer labs with a server that connects these labs in a parallel configuration under the 2-out-of-3: G policy. For reliability and availability enhancement, we applied random processes using supplementary variables and a neural network approach. In the proposed complex system, failure can occur because of at least two computer lab failures and one server failure or because of some cataclysmic failure. The unit failure rates are fixed (constant) and assumed to be exponentially distributed. The computer lab network was maintained using two types of repair. The repairs that restore partial or non-fatal failures to the original condition are general or Type I repairs, whereas Type II or copula repairs are used to restore complete failures. All failure and repair rates were considered as neural weights when analyzed using the neural network approach governed by an exponential distribution. The probabilities of system states, up and down state probabilities, availability, and reliability of the model are evaluated using a Markovian process and Laplace transformations. In summary, the results obtained using MAPLE software and MATLAB software are presented in tables and graphs.

The remainder of this paper is structured as follows: Section 2 presents a comprehensive system description, including the assumptions and notations pertinent to the model under consideration. Section 3 offers a comprehensive elucidation of the state transition diagram, encompassing a description of each state. Section 4 delineates the partial differential equations employing the supplementary variable technique in accordance with the state transition diagram, initial and boundary conditions, and their solutions facilitated by Laplace transformation. The ANN technique, encompassing both the algorithm and the solution methodology, is described in Section 5. The solution utilizing MAPLE software and MATLAB software for copula repair and the ANN is presented in Section 6. Finally, Section 7 presents an analysis of the results, together with conclusions and key insights.

2.2 Assumptions

We consider the following assumptions for the lab network:

1) A set of three computer labs is connected through a server in a parallel configuration under a 2-out-of-3 G policy that utilizes a random process.
2) Initially, it begins in state S_0, in which every lab is operational and functions correctly.
3) The system continues to operate; however, any failure in the lab lowers the effectiveness of the system.
4) A maintenance technician is on-site and can be called upon whenever the system is in a partially or entirely failed state.
5) The system operates effectively after repair. Repair of the system did not result in any damage.
6) Every failure rate assumes a constant value and follows exponential distribution.
7) Partially failed systems are remedied using a general distribution, whereas fully failed systems are addressed using the copula distribution from the Gumbel-Hougaard family.

For clarity, the following assumptions are adopted throughout the analysis: (1) the system is initialized in the fully operational (all-up) state at $t = 0$; hence $P_{up}(0) = 1$ and $R(0) = 1$; (2) failures and repairs are modeled as exponentially distributed with constant rates, which leads to a continuous-time Markov representation; (3) transition rates are treated as independent parameters as commonly assumed in repairable system modeling; and (4) state probabilities and derived performance measures (availability/reliability) are bounded within $[0,1]$. We verified these conditions numerically and ensured that the reported tables and figures satisfied them.

2.3 Notations

The following notations are used throughout the manuscript:

n Total number of components in the system

k Minimum number of components that must function correctly for an operational system.

x Supplementary variable / elapsed time for repair

S Represent the states.

C Catastrophic or cataclysmic failure

s, t Laplace transform / Time scale variables

$\alpha / \beta / \gamma$ Failure rate of computer lab-1/ lab-2/ lab-3.

λ_S Failure rate due to server failure.

λ_C Catastrophic failure rate.

$\phi(x)$ Repair rate of computer lab-1/ lab-2/ lab-3.

$\mu_0(x)$ Repair rate while the system is in complete failure mode.

$P_0(t)$ The state transition probability that the system is in S_i at an instant for $i = 0$.

$\bar{P}(s)$ Laplace transformation of the state transition probability $P(t)$.

$P_i(x,t)$ The Probability that the system is in state S_i for $i = 1$ to 10 , where x is a repaired variable and t is a time variable.

K_1, K_2 Revenue generated and service cost per unit time respectively.

$\mu_0(x)$ Repair rate when the system fails under the Gumbel-Hougaard family copula $\mu_0(x) = \exp\left[x^\theta + \{\log\phi(x)\}^\theta\right]^{1/\theta}$ where, $u_1(x) = \phi(x)$ and $u_2(x) = e^x$.

With the system description, assumptions, and notation established, the next section formulates the stochastic model and derives the governing equations required for the reliability and availability evaluation.

3. State Transition Diagram

A transition diagram for the various states is shown in **Figure 1**. In this figure, S_0 is the initial and perfect state; S_1, S_3 and S_5 are the partially degraded states; and S_2, S_4, S_6, S_7 and S_8 are fully non-functional states. If the first lab fails, the system moves to S_1; if the second lab fails to move to S_3 and if the third lab fails, it moves to S_5. It is also noted here that, owing to the failure of the first and second labs, the system approaches the S_2 state, the failure of the first and third labs to S_6 state, and the failure of the second and third labs will move to S_4 state. Moreover, states S_7 and S_8 have completely failed states owing to server and catastrophic failures.

Based on the derived transition structure and governing relations, we now specify the boundary/initial conditions and outline the analytical solution procedure used to compute state probabilities.

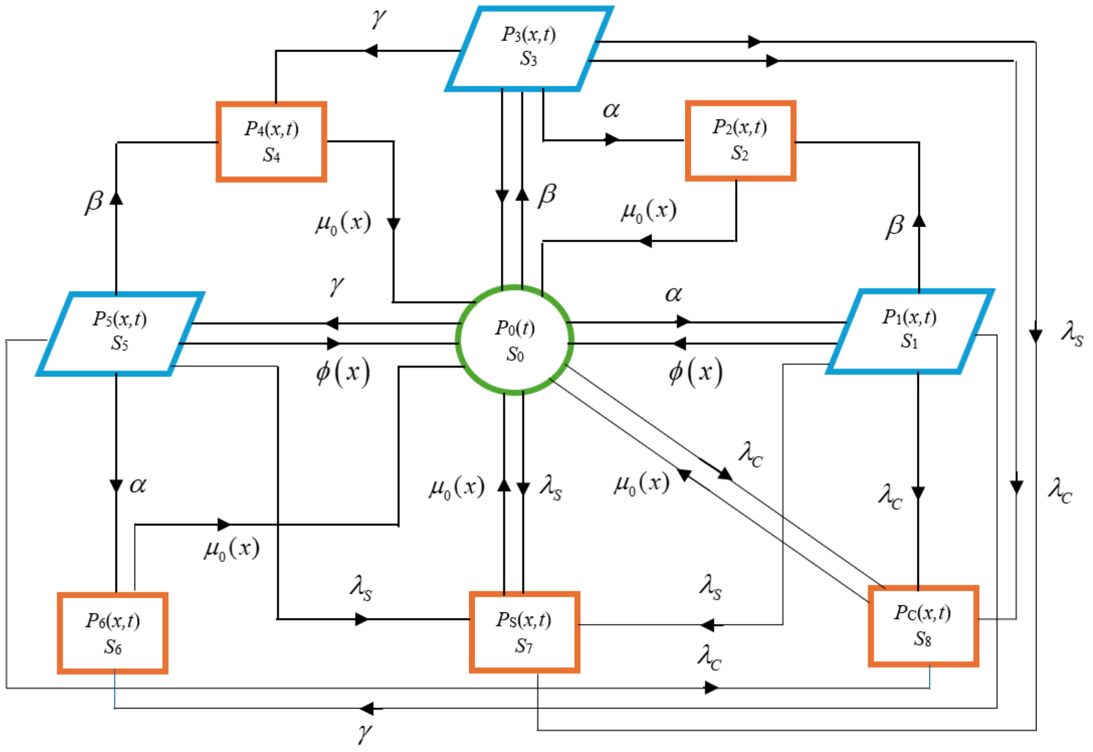

Figure 1. State transition diagram of the model.

4. Mathematical Formulation

From the probability of consideration and continuity arguments, we obtain the following differential equations linked to the current mathematical model. In all equations, x (elapsed repair time) and t (schedule time) represent time. The supplementary variable x varies when the system is in a degraded or completely failed state, and its rate of variation is equal to t.

$$\left[\frac{\partial}{\partial t}+\alpha+\beta+\gamma+\lambda_S+\lambda_C\right]P_0(t)=\int_0^\infty \phi(x)P_1(x,t)\,dx+\int_0^\infty \phi(x)P_3(x,t)\,dx+\int_0^\infty \phi(x)P_5(x,t)\,dx$$

$$+\sum_k\int_0^\infty \exp\left[x^\theta+\{\log\phi(x)\}^\theta\right]^{1/\theta}P_k(x,t)\,dx\{k=2,4,6,S,C\} \tag{1}$$

$$\left[\frac{\partial}{\partial t}+\frac{\partial}{\partial x}+\beta+\gamma+\lambda_S+\lambda_C+\phi(x)\right]P_1(x,t)=0 \tag{2}$$

$$\left[\frac{\partial}{\partial t}+\frac{\partial}{\partial x}+\exp\left[x^\theta+\{\log\phi(x)\}^\theta\right]^{1/\theta}\right]P_2(x,t)=0 \tag{3}$$

$$\left[\frac{\partial}{\partial t}+\frac{\partial}{\partial x}+\alpha+\gamma+\lambda_S+\lambda_C+\phi(x)\right]P_3(x,t)=0 \tag{4}$$

$$\left[\frac{\partial}{\partial t}+\frac{\partial}{\partial x}+\exp\left[x^{\theta}+\left\{\log\phi\left(x\right)\right\}^{\theta}\right]^{1/\theta}\right]P_4\left(x,t\right)=0 \tag{5}$$

$$\left[\frac{\partial}{\partial t}+\frac{\partial}{\partial x}+\alpha+\beta+\lambda_S+\lambda_C+\phi\left(x\right)\right]P_5\left(x,t\right)=0 \tag{6}$$

$$\left[\frac{\partial}{\partial t}+\frac{\partial}{\partial x}+\exp\left[x^{\theta}+\left\{\log\phi\left(x\right)\right\}^{\theta}\right]^{1/\theta}\right]P_6\left(x,t\right)=0 \tag{7}$$

$$\left[\frac{\partial}{\partial t}+\frac{\partial}{\partial x}+\exp\left[x^{\theta}+\left\{\log\phi\left(x\right)\right\}^{\theta}\right]^{1/\theta}\right]P_S\left(x,t\right)=0 \tag{8}$$

$$\left[\frac{\partial}{\partial t}+\frac{\partial}{\partial x}+\exp\left[x^{\theta}+\left\{\log\phi\left(x\right)\right\}^{\theta}\right]^{1/\theta}\right]P_C\left(x,t\right)=0 \tag{9}$$

4.1 Boundary Conditions of the Model

$$P_1\left(0,t\right)=\alpha P_0\left(t\right) \tag{10}$$

$$P_2\left(0,t\right)=\beta P_1\left(0,t\right)+\alpha P_3\left(0,t\right)=2\alpha\beta P_0\left(t\right) \tag{11}$$

$$P_3\left(0,t\right)=\beta P_0\left(t\right) \tag{12}$$

$$P_4\left(0,t\right)=\gamma P_3\left(0,t\right)+\beta P_5\left(0,t\right)=2\beta\gamma P_0\left(t\right) \tag{13}$$

$$P_5\left(0,t\right)=\gamma P_0\left(t\right) \tag{14}$$

$$P_5\left(0,t\right)=\gamma P_1\left(0,t\right)+\alpha P_5\left(0,t\right)=2\alpha\gamma P_0\left(t\right) \tag{15}$$

$$P_S\left(0,t\right)=\lambda_S\left[P_0\left(t\right)+P_1\left(0,t\right)+P_3\left(0,t\right)+P_5\left(0,t\right)\right]=\lambda_S\left(\alpha+\beta+\gamma\right)P_0\left(t\right) \tag{16}$$

$$P_C\left(0,t\right)=\lambda_C\left[P_0\left(t\right)+P_1\left(0,t\right)+P_3\left(0,t\right)+P_5\left(0,t\right)\right]=\lambda_C\left(\alpha+\beta+\gamma\right)P_0\left(t\right) \tag{17}$$

4.2 Initial Conditions of the Model

$P_0\left(0\right)=1$, and the other state probabilities are zero at $t=0$, that is $P_i\left(x,0\right)=0\ \forall\ \ i=1$ to $6,S,C$

4.3 Solution of the Model

To solve Equations (1) to (9) with the help of boundary conditions from Equations (10) to (17) and initial conditions, let us take the Laplace transform. We obtain

$$\left[s+\alpha+\beta+\gamma+\lambda_S+\lambda_C\right]\bar{P}_0(s)=1+\int_0^\infty\phi(x)\bar{P}_1(x,s)\,dx+\int_0^\infty\phi(x)\bar{P}_3(x,s)\,dx+\int_0^\infty\phi(x)\bar{P}_5(x,s)\,dx$$

$$+\sum_k\int_0^\infty\exp\left[x^\theta+\left\{\log\phi(x)\right\}^\theta\right]^{1/\theta}\bar{P}_k(x,s)\,dx\left\{k=2,4,6,S,C\right\}\tag{18}$$

$$\left[s+\frac{\partial}{\partial x}+\beta+\gamma+\lambda_S+\lambda_C+\phi(x)\right]\bar{P}_1(x,s)=0\tag{19}$$

$$\left[s+\frac{\partial}{\partial x}+\exp\left[x^\theta+\left\{\log\phi(x)\right\}^\theta\right]^{1/\theta}\right]\bar{P}_2(x,s)=0\tag{20}$$

$$\left[s+\frac{\partial}{\partial x}+\alpha+\gamma+\lambda_S+\lambda_C+\phi(x)\right]\bar{P}_3(x,s)=0\tag{21}$$

$$\left[s+\frac{\partial}{\partial x}+\exp\left[x^\theta+\left\{\log\phi(x)\right\}^\theta\right]^{1/\theta}\right]\bar{P}_4(x,s)=0\tag{22}$$

$$\left[s+\frac{\partial}{\partial x}+\alpha+\beta+\lambda_S+\lambda_C+\phi(x)\right]\bar{P}_5(x,s)=0\tag{23}$$

$$\left[s+\frac{\partial}{\partial x}+\exp\left[x^\theta+\left\{\log\phi(x)\right\}^\theta\right]^{1/\theta}\right]\bar{P}_6(x,s)=0\tag{24}$$

$$\left[s+\frac{\partial}{\partial x}+\exp\left[x^\theta+\left\{\log\phi(x)\right\}^\theta\right]^{1/\theta}\right]\bar{P}_S(x,s)=0\tag{25}$$

$$\left[s+\frac{\partial}{\partial x}+\exp\left[x^\theta+\left\{\log\phi(x)\right\}^\theta\right]^{1/\theta}\right]\bar{P}_C(x,s)=0\tag{26}$$

Boundary conditions

$$\bar{P}_1(0,s)=\alpha\bar{P}_0(s)\tag{27}$$

$$\bar{P}_2(0,s)=2\alpha\beta\bar{P}_0(s)\tag{28}$$

$$\bar{P}_3(0,s)=\beta\bar{P}_0(s)\tag{29}$$

$$\bar{P}_4(0,s)=2\beta\gamma\bar{P}_0(s)\tag{30}$$

$$\bar{P}_5(0,s)=\gamma\bar{P}_0(s)\tag{31}$$

$$\bar{P}_6(0,s)=2\alpha\gamma\bar{P}_0(s)\tag{32}$$

$$\bar{P}_S\left(0,s\right)=\lambda_S\left(\alpha+\beta+\gamma\right)\bar{P}_0\left(s\right) \tag{33}$$

$$\bar{P}_C\left(0,s\right)=\lambda_C\left(\alpha+\beta+\gamma\right)\bar{P}_0\left(s\right) \tag{34}$$

Solve Equations (18) - (26) with the help of Equations (27) - (34) and use the first shifting theorem of the Laplace transform

$$\int_0^\infty \phi\left(x\right)e^{-sx-\int_0^\infty \phi(x)dx}dx=\bar{S}_\phi\left(x\right)=\frac{\phi}{s+\phi}\text{ and }\int_0^\infty e^{-sx-\int_0^\infty \phi(x)dx}dx=\frac{1-\bar{S}_\phi\left(x\right)}{s}=\frac{1}{s+\phi} \tag{35}$$

One may get the following outputs

$$\bar{P}_0\left(s\right)=\frac{1}{D\left(s\right)} \tag{36}$$

$$\bar{P}_1\left(s\right)=\frac{\alpha}{D\left(s\right)}\frac{1}{s+\beta+\gamma+\lambda_S+\lambda_C+\phi\left(x\right)} \tag{37}$$

$$\bar{P}_2\left(s\right)=\frac{2\alpha\beta}{D\left(s\right)}\frac{1}{s+\mu_0\left(x\right)} \tag{38}$$

$$\bar{P}_3\left(s\right)=\frac{\beta}{D\left(s\right)}\frac{1}{s+\gamma+\alpha+\lambda_S+\lambda_C+\phi\left(x\right)} \tag{39}$$

$$\bar{P}_4\left(s\right)=\frac{2\beta\gamma}{D\left(s\right)}\frac{1}{s+\mu_0\left(x\right)} \tag{40}$$

$$\bar{P}_5\left(s\right)=\frac{\gamma}{D\left(s\right)}\frac{1}{s+\alpha+\beta+\lambda_S+\lambda_C+\phi\left(x\right)} \tag{41}$$

$$\bar{P}_6\left(s\right)=\frac{2\alpha\gamma}{D\left(s\right)}\frac{1}{s+\mu_0\left(x\right)} \tag{42}$$

$$\bar{P}_S\left(s\right)=\frac{\lambda_S\left(\alpha+\beta+\gamma\right)}{D\left(s\right)}\frac{1}{s+\mu_0\left(x\right)} \tag{43}$$

$$\bar{P}_C\left(s\right)=\frac{\lambda_C\left(\alpha+\beta+\gamma\right)}{D\left(s\right)}\frac{1}{s+\mu_0\left(x\right)} \tag{44}$$

where,

$$D\left(s\right)=s+\alpha+\beta+\gamma+\lambda_S+\lambda_C-\frac{\alpha\phi}{s+\beta+\gamma+\lambda_S+\lambda_C+\phi}-\frac{\beta\phi}{s+\alpha+\gamma+\lambda_S+\lambda_C+\phi}$$

$$-\frac{\gamma\phi}{s+\alpha+\beta+\lambda_S+\lambda_C+\phi}-\left\{2\left(\alpha\beta+\beta\gamma+\alpha\gamma\right)+\left(\alpha+\beta+\gamma\right)\left(\lambda_S+\lambda_C\right)\right\}\frac{\mu_0}{s+\mu_0}$$

The probabilities in the up state(s) (when the system is operating) and down state(s) (when the system is failing) can be expressed as

$$\bar{P}_{up}\left(s\right)=\frac{1}{D\left(s\right)}\begin{bmatrix}1+\dfrac{\alpha}{s+\beta+\gamma+\lambda_S+\lambda_C+\phi\left(x\right)}+\dfrac{\beta}{s+\gamma+\alpha+\lambda_S+\lambda_C+\phi\left(x\right)}\\+\dfrac{\gamma}{s+\alpha+\beta+\lambda_S+\lambda_C+\phi\left(x\right)}\end{bmatrix} \tag{45}$$

$$\bar{P}_{down}\left(s\right)=1-\bar{P}_{up}\left(s\right) \tag{46}$$

It should be noted that from Equations (45) and (46), respectively: $\bar{P}_{up}\left(s\right)+\bar{P}_{down}\left(s\right)={}^{1}\!/_{s}$.

Furthermore, if the repair follows

(i) General distribution $\bar{S}_{\phi}\left(s\right)=\dfrac{\phi}{s+\phi}$.

(ii) Gumbel-Hougaard Family Copula Distribution $\bar{S}_{\mu_0}\left(s\right)=\dfrac{\exp\left[x^{\theta}+\left\{\log\phi\left(x\right)\right\}^{\theta}\right]^{1/\theta}}{s+\exp\left[x^{\theta}+\left\{\log\phi\left(x\right)\right\}^{\theta}\right]^{1/\theta}}$.

These analytical expressions provide the basis for computing time-dependent state probabilities and, consequently, the reliability and availability measures used in the numerical section.

5. ANN (Artificial Neural Network) Technique

An ANN, or artificial neural network, is a computationally based model that derives its idea from a biological neural network. It functions as a human artificial nervous system that receives, processes, and transmits signals from a source. The network has three layers: an input layer, output layer, and hidden layer. The number of neurons in the input and hidden layers was equal to the number of states in the transition diagram. The failure and repair rates serve as the neural weights. There are various algorithms for training neural networks, but we chose the backpropagation method for training the neural network. A neural network has the desired reliability corresponding to its weight. We solved this problem using the designed neural networks and MATLAB software. The programming language feed-forward was applied during training. There are no cycles or loops in a feed-forward neural network. In this network, information flows in one direction, starting from the input layer and moving first to the hidden layer and then to the output layer. The ANN model is illustrated in **Figure 2**.

5.1 Algorithm for ANN of the System

To estimate the analytically obtained availability and reliability measures, a feedforward ANN was trained using a standard backpropagation procedure. The computational steps are summarized in Algorithm 1.

Algorithm 1: Feedforward ANN training for availability/reliability estimation

```
Require: Training set $\{(u\_k, t\_k)\}\_\{k = 1..N\}$, hidden neurons $n_h$, learning rate η, tolerance ε, max epochs E_max
Ensure: Trained weights W (input→hidden), V (hidden→output), biases b_h, b_o

// Step 1: Define network size
Set $n_i n = dimension(u_k), n_o ut = dimension(t_k)$

// Step 2: Initialize parameters
Initialize $W \in R\^\{n\_h \times n\_in\}$, $V \in R\^\{n\_out \times n\_h\}$ with small random values
Initialize $b\_h \in R\^\{n\_h\}$, $b\_o \in R\^\{n\_out\}$ with small random values

    for epoch = 1 to E_max do
    SSE = 0

    for k = 1 to N do
    // Forward pass
    $z = Wu_k + b_h$
    $h = logsig(z)$
    $s = Vh + b_o$
    $y = logsig(s)$

    // Error
    $e = t_k - y$
    $SSE = SSE + ||e||^2$

    // Backpropagation (logsig derivative: a(1−a))
    $\delta\_o = (e) \odot y \odot (1 - y)$
    $\delta_h = (V^T \delta_o) \odot h \odot (1 - h)$

    // Weight and bias updates
    $V = V + \eta(\delta_o h^T)$
    $b_o = b_o + \eta\delta_o$
    $W = W + \eta\left(\delta_h u_k^T\right)$
    $b_h = b_h + \eta\delta_h$
    end for

    $MSE = \frac{SSE}{N}$
    if $MSE \leq \varepsilon$ then
    break
    end if
    end for

    Return $W, V, b_h, b_o$
```

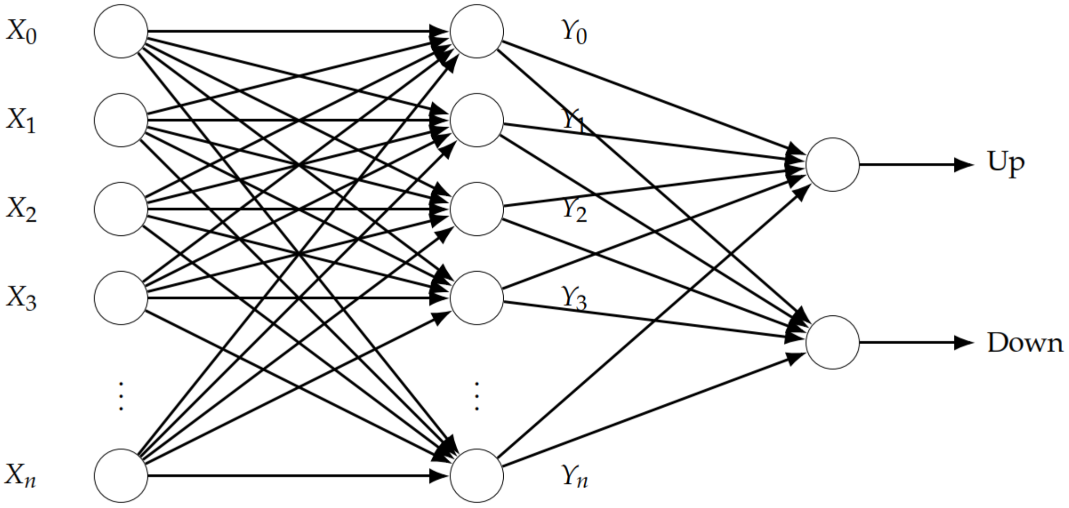

Figure 2. ANN model.

5.2 Solution of the Model using ANN

Let $\mathbf{P}(\boldsymbol{t}) = [\boldsymbol{P_0}(\boldsymbol{t}), \boldsymbol{P_1}(\boldsymbol{t}), \dots, \boldsymbol{P_8}(\boldsymbol{t})]^{\boldsymbol{T}}$denote the state-probability vector of the continuous-time Markov model associated with the state-transition diagram (**Figure 1**). The elements $\boldsymbol{P_i}(\boldsymbol{t})$represent the probability that the system is in state $\boldsymbol{S_i}$ at time $\boldsymbol{t}$, and satisfy the Kolmogorov forward (master) equations governed by the failure and repair rates. To approximate these state probabilities using a supervised feedforward ANN, we assigned one output neuron per state and defined the ANN output as the corresponding state probability. Therefore, the ANN output $\boldsymbol{Y_i}(\boldsymbol{t})$is considered as $\boldsymbol{P_i}(\boldsymbol{t})$, which directly leads to Equation (47).

For training and implementation, time was sampled on a uniform grid $\boldsymbol{t_k} = \boldsymbol{k\Delta t}$. We form input–target pairs as $\mathbf{X}(\boldsymbol{t_k}) = \mathbf{P}(\boldsymbol{t_k})$ and $\mathbf{Y}(\boldsymbol{t_k}) = \mathbf{P}(\boldsymbol{t_{k+1}})$. Using a one-step (first-order) discretization of the forward equations, $\mathbf{P}(\boldsymbol{t_{k+1}})$ can be approximately expressed as a weighted combination of $\mathbf{P}(\boldsymbol{t_k})$, which motivates the weighted-sum relations used in the subsequent equations. Nonzero interconnections (weights) are selected according to the admissible transitions shown in **Figure 1**.

We can write the inputs for the model as

$$Y_i = P_i(t) \tag{47}$$

where, $i = 0,1,2,3,\dots,8$ indicates the position of the state in the transition diagram of the mode. We can now generate the equations for the neural network of the model as

$$Y_i = P_i(t + \Delta t), \text{ where } i = 0,1,2,3,\dots,8 \tag{48}$$

$$Y_0 = w_{00}X_0 + w_{10}X_1 + w_{20}X_2 + w_{30}X_3 + w_{40}X_4 + w_{50}X_5 + w_{60}X_6 + w_{70}X_7 + w_{80}X_8 \tag{49}$$

$$Y_1 = w_{01}X_0 + w_{11}X_1 \tag{50}$$

$$Y_2 = w_{12}X_1 + w_{22}X_2 + w_{32}X_3 \tag{51}$$

$$Y_3 = w_{03}X_0 + w_{33}X_3 \tag{52}$$

$$Y_4 = w_{34}X_3 + w_{44}X_4 + w_{54}X_5 \tag{53}$$

$$Y_5 = w_{05}X_0 + w_{55}X_5 \tag{54}$$

$$Y_6 = w_{16}X_1 + w_{56}X_5 + w_{66}X_6 \tag{55}$$

$$Y_7 = w_{07}X_0 + w_{17}X_1 + w_{37}X_3 + w_{57}X_5 + w_{77}X_7 \tag{56}$$

$$Y_8 = w_{08}X_0 + w_{18}X_1 + w_{38}X_3 + w_{58}X_5 + w_{88}X_8 \tag{57}$$

The upward and downward state's probabilities may be given as follows

$$P_{UP}\left(t\right) = Y_0 + Y_1 + Y_3 + Y_5 \tag{58}$$

$$P_{DOWN}\left(t\right) = 1 - P_{UP}\left(t\right) \tag{59}$$

Finally, the reliability of the ANN model computed as follows

$$\text{Reliability} = Y_0 + Y_1 + Y_3 + Y_5 \tag{60}$$

In the next section, the analytically computed reliability/availability results are used as reference data to validate ANN-based estimates through numerical comparisons.

6. Results and Discussions

6.1 Analytical Availability Analysis of the Model

To compute availability using the supplementary variable technique and the Gumbel-Hougaard family copula distribution, we implemented Equation (45) while choosing various parameters such as $\alpha = 0.10$, $\beta = 0.11, \gamma = 0.12, \lambda_S = 0.15, \lambda_C = 0.16, \phi = 1, x = 1$ and using MAPLE software for the inverse Laplace transform, we obtain the availability as a function of time t as

$$\begin{aligned} P_{up}(t) = {} & 0.007589e^{-1.0520t} + 0.000114e^{-2.7182t} + 0.051684e^{-1.2035t} + 4.7924\ 10^{-6}e^{-1.1435t} \\ & + 4.7002\ 10^{-6}e^{-1.1424t} + 0.953776e^{-0.1031t} - 0.006919e^{-1.0530t} - 0.006255e^{-1.0540t} \end{aligned} \tag{61}$$

Table 1. Availability $P_{up}(t)$ with respect to time using MAPLE software.

Time	$P_{up}(t)$
0	1.0000
1	0.8738
2	0.7799
3	0.7011
4	0.6317
5	0.5695
6	0.5137
7	0.4633
8	0.4179
9	0.3769
10	0.3400

Taking a time span of 10 units of time t, the values of availability are depicted in **Table 1**, which represents the relationship between availability $P_{up}(t)$ and time t.

Having obtained the analytical availability profile, we evaluated the corresponding analytical reliability using the same modeling framework.

6.2 Analytical Reliability Analysis of the Model

Considering Equation (45) in the context of the supplementary variable technique and the Gumbel-Hougaard family copula distribution, to calculate reliability, all parameters are kept constant, as in the case of availability without repair. After fixing all parameters, we apply an inverse Laplace transformation using MAPLE software and obtain the following expression for reliability:

$$R(t) = -0.107142e^{-0.0520t} + 0.099099e^{-0.0530t} + 0.090909e^{-0.0540t} + 0.917134e^{-0.1640t} \tag{62}$$

Taking a time span of 10 units of time t, the values of reliability are shown in **Table 2** which represents the relationship between reliability $R(t)$ and time t.

Table 2. Reliability $R(t)$ with respect to time using MAPLE software.

Time	$R(t)$
0	1.0000
1	0.8568
2	0.7348
3	0.6309
4	0.5423
5	0.4667
6	0.4022
7	0.3472
8	0.3001
9	0.2599
10	0.2255

These analytical curves served as reference results for assessing the accuracy of the ANN-based estimates.

6.3 Comparison $P_{up}(t)$ Vs $R(t)$ with Respect to Time

Figure 3 provides a visual exploration of how system availability and reliability evolve over a time span of ten units, using eight distinct plotting techniques, each offering unique insight into operational trends. A traditional line plot served as the initial representation, illustrating the decline in both metrics over time. Availability begins at a maximum value of 1.0000 and gradually decreases to 0.3613, whereas reliability exhibits a more pronounced drop from 1.0000 to 0.2235. The consistent dominance of availability over reliability throughout the timeline reflects the system's ability to recover or continue functioning in the presence of failures, likely due to built-in repair mechanisms or redundant configurations. This form of representation is particularly helpful in summarizing system conditions and performance boundaries within a two-dimensional frame, and may assist in the identification of critical intervention thresholds.

An area plot was constructed to assess the cumulative impact of this degradation. This visualization highlights the total span during which the system remains operational versus failure-free. The area under the availability curve was visibly greater than that of the reliability curve, indicating longer periods of functional uptime. For example, at $t = 5$, the availability stands at 0.5724, whereas the reliability decreased to 0.4666. Such a contrast reinforces the interpretation that the availability metric accounts for recovery dynamics, whereas reliability reflects uninterrupted operations without repair.

A step plot was used to represent systems that were monitored at discrete intervals. This plot captures the staircase-like behavior of both metrics, where the values remain static for a period before dropping at specific points. These transitions emulate practical systems that undergo routine inspections or status updates at scheduled times. Notably, between $t = 3$ and $t = 4$, availability falls from 0.7011 to 0.6317, while reliability decreased from 0.6309 to 0.5423. This emphasizes the episodic nature of the degradation events captured in the monitored environments.

In addition, a dual-axis plot was implemented to simultaneously present both metrics using separate Y-axes while sharing a common time axis. This arrangement preserves the scale and resolution of each measure, making it easier to interpret their respective behaviors. The availability data are plotted against the left axis, and the reliability data against the right. This layout reveals not only the distinct decay rates but also the differing patterns of decline, offering a clearer perspective on how system resilience and failure rates evolve in tandem. Such visualization is particularly helpful when comparing metrics that respond to separate operational processes but ultimately influences overall system dependability.

The temporal profiles of the system availability and reliability were further examined using a series of advanced graphical representations, as illustrated in the last four pictures of **Figure 3**. Each plot was selected to reveal its specific behavioral patterns and complementary perspectives on the degradation process. A semi-logarithmic plot was first utilized to highlight the exponential decay of both metrics over time. Availability started at 1.0000 and gradually decreased to 0.3613 at $t = 10$, whereas reliability showed a more pronounced decline from 1.0000 to 0.2235 within the same period. By applying a logarithmic scale to the Y-axis, the early-time sensitivity of reliability is more clearly exposed, indicating that system failures tend to accumulate rapidly in the initial phase. In contrast, availability remains comparatively elevated due to the presence of mechanisms such as repair actions, system resets, or redundancy pathways that help preserve uptime despite failures.

To examine how these two measures diverge during the system operation, a difference plot was generated by computing the pointwise subtraction of reliability from availability. The largest observed gap occurred at $t = 9$, where the availability was 0.3950 and the reliability was 0.2583, yielding a difference of approximately 0.0789. This consistent separation across the time domain suggests that the system retains a certain capacity to remain operational even as its failure-free performance diminishes. Such a visualization proves valuable in assessing how effectively the system mitigates the consequences of failure through recovery strategies.

A gradient-based plot was also introduced to evaluate the rate at which the availability and reliability change with respect to time. The initial interval from $t = 0$ to $t = 1$ shows a steeper descent in reliability, with a drop of approximately -0.1432, compared to -0.1262 for availability. Over subsequent time steps, the gradient values tended to flatten, indicating predictable slowing of the degradation process. This analysis offers key insights into the periods of accelerated wear or stress on the system and supports the formulation of timely maintenance schedules.

After validating the analytical trends, we proceeded with ANN-based availability estimation and its agreement with the analytical results.

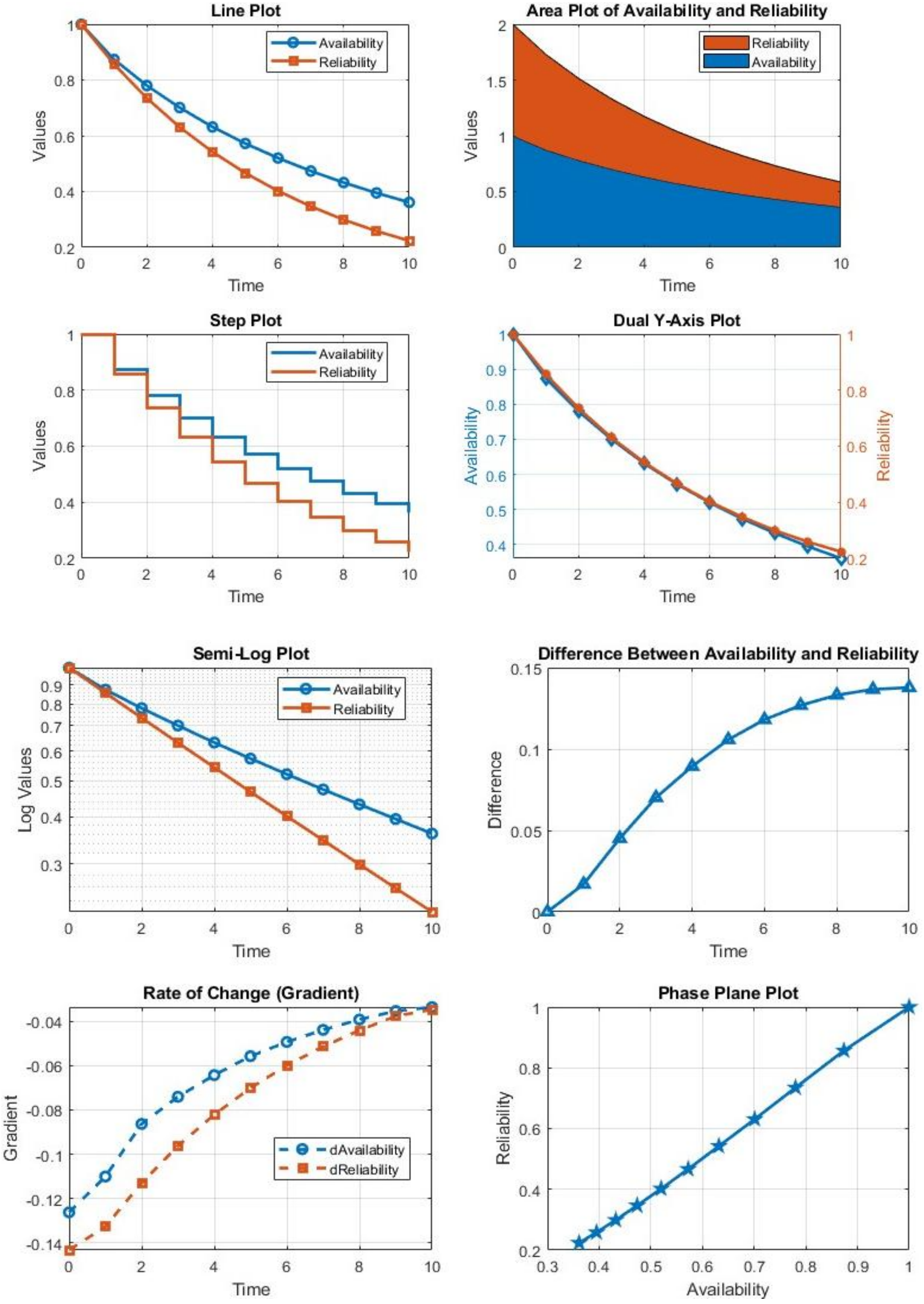

Figure 3. Comparison $P_{up}(t)$ Vs $R(t)$ with respect to time.

6.4 ANN Availability Analysis of the Model

Although analytical expressions for reliability and availability can be derived for the considered model, practical evaluation often requires repeated computations under multiple parameter settings for sensitivity analysis and design comparison. In such scenarios, repeatedly solving the analytical expressions becomes time-consuming and cumbersome. Therefore, we employ the ANN as a fast surrogate estimator trained on analytically generated data, which can approximate the reliability/availability curves with very low computational cost, enabling rapid prediction for new time points and parameter combinations within the studied range. Moreover, the same ANN-based estimation framework can be extended to more complex repairable network models, in which closed-form solutions may be difficult to obtain.

A complex exponential function was used to analytically model the availability function, and its accuracy was compared to that of an ANN approximation developed in MATLAB software. With only three hidden neurons, the ANN achieved a high precision in replicating the analytical availability values across the time period from 0 to 10 times units. The absolute difference between the ANN predictions and analytical values was small, with a maximum absolute error of approximately 3.63×10^{-2}at the initial time point. The mean squared error (MSE) was 1.199×10^{-4}, indicating close agreement between the two models. These findings demonstrate the ability of ANN to effectively learn the underlying pattern of the availability function, even with a simplified network configuration. **Figure 4** displays a comparison between the analytical and ANN predicted availability values, highlighting the ANN's close alignment with the analytical curve throughout the time domain. **Table 3** presents the ANN-predicted availability values. At $t = 0$, the system is assumed to start in a fully operational (all-up) state. Therefore, the initial condition was $P_{up}(0) = 1$, and the corresponding entry in **Table 3** is 1.0000. In the ANN implementation, this initial condition was enforced by including the point $t = 0$ with the target $P_{up}(0) = 1$in the training/validation dataset so that the ANN prediction was consistent with the analytical model at the initial time.

Table 3. Availability $P_{up}(t)$ with respect to time using ANN.

Time	$P_{up}(t)$ using ANN
0	0.8644
1	0.8738
2	0.7799
3	0.7011
4	0.6317
5	0.5695
6	0.5137
7	0.4633
8	0.4179
9	0.3769
10	0.3400

Furthermore, the network was initialized to satisfy the boundary/initial condition $A(0) = 1$, and the ANN outputs were constrained to remain within $[0,1]$ by using a logistic activation function. To further enhance interpretability, the ANN-estimated availability values are reported alongside the analytically obtained Markov/supplementary variable results at identical time points in **Table 3**, enabling the direct verification of the estimation accuracy.

To ensure effective learning from the actual available data, only the weight initialization from the scaled pretraining phase was retained, whereas the network itself was freshly configured for the final training. The ANN was trained for 200 epochs, during which the prediction of $A(0)$ was monitored at each step. As shown in **Table 4** and **Figure 4**, the model began with an initial output of $A(0) = 0.7840$ and steadily improved, reaching 0.9414 at epoch 5, 0.9684 at epoch 50, and converged to 0.9797 at epoch 200. Although the model did not achieve the exact target of $A(0) = 1.0000$ within the given training limit (with a tolerance threshold of 10^{-4}), it demonstrated a highly stable and smooth convergence pattern. This confirms the effectiveness of the ANN in approximating the availability behavior of the system under study.

Table 4. Availability $P_{up}(t)$ with respect to number of epochs.

Iteration	$P_{up}(t)$
1	0.7840
5	0.9414
10	0.9583
20	0.9648
35	0.9670
52	0.9686
100	0.9725
150	0.9765
200	0.9767

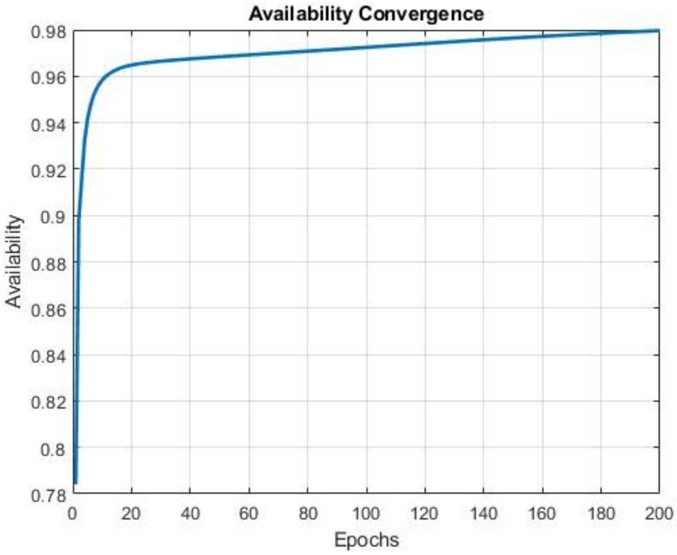

Figure 4. Availability $P_{up}(t)$ convergence with respect to number of epochs.

In parallel with availability estimation, we trained and evaluated the ANN model for reliability estimation under the same network architecture and training settings.

6.5 ANN Reliability Analysis of the Model

The reliability function is estimated using the ANN model in MATLAB software. The fitting of the network, which contained three hidden neurons, was carried out over a time span from 0 to 10 times units. The findings revealed that the ANN output was highly consistent with the analytical values, and there was little error in most time intervals. At the starting point, the maximum absolute error was 0.0514. Interestingly, this value was much lower at other time points, usually around 10^{-4} or lower. The MSE value obtained was 0.0002414, suggesting a strong relationship between the model predictions and the outputs of the analytical model. **Figure 5** presents side-by-side graphs of the two models, visually portraying the differences between the analyzed and ANN models concerning reliability and showing how the ANN can be trained to understand the dynamics of the reliability function. The ANN predicted availability values for reliability are given in **Table 5**, for which reliability is shown in accessibility forms. Similarly, reliability is defined by the standard initial condition, in which the system is failure-free at $t = 0$. Hence, $R(0) = 1$, and the $t = 0$ entry in **Table 5** was corrected to 1.0000.

The ANN predictions were compared directly with the analytical results at all reported time points (**Tables 3** and **5**), demonstrating close agreement and validating the ANN as an estimator of the analytical reliability/availability curves. Similarly, **Table 5** presents the analytical reliability values from the Markov/supplementary-variable model and the corresponding ANN estimates at the same time points.

Table 5. Reliability $R(t)$ with respect to time using ANN.

Time	$R(t)$ using ANN
0	0.94857
1	0.8568
2	0.7348
3	0.6309
4	0.5423
5	0.4667
6	0.4022
7	0.3472
8	0.3001
9	0.2599
10	0.2255

Moreover, the network was initialized with a starting prediction near R(0)≈0.6 and trained over 200 epochs. Its structure comprises a single hidden layer with 10 neurons, where both the hidden and output layers use the logsig activation function to ensure that the outputs remain bounded between 0 and 1. Throughout the training process, the ANN exhibited steady improvement in approximating the target values. As shown in **Table 6** and **Figure 5**, the model began with a predicted reliability of $R(0) = 0.6141$ at epoch 1, the network reached $R(0) = 0.6206$ by epoch 5, and reached a significant leap to $R(0) = 0.9015$ by epoch 10. The learning continued with a value of $R(0) = 0.9361$ at epoch 20, and further refinements were observed as the prediction improved to $R(0) = 0.9529$ at epoch 100, $R(0) = 0.9814$ at epoch 150, and finally $R(0) = 0.9849$ at epoch 200. Although the model did not reach the exact target of $R(0) = 1.0000$ within the training window (considering a tolerance of 10^{-4}), the results affirmed the ANN's ability to accurately learn and approximate the system's reliability behavior with high precision.

Table 6. Reliability $R(t)$ with respect to number of epochs.

Iteration	$R(t)$
1	0.6141
5	0.6206
10	0.9015
20	0.9361
35	0.9386
52	0.9401
100	0.9529
150	0.9814
200	0.9849

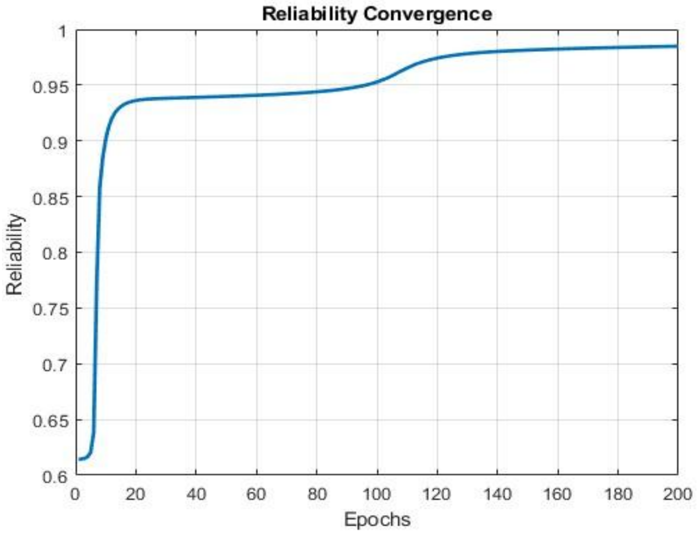

Figure 5. Reliability $R(t)$ convergence with respect to number of epochs.

Finally, a direct comparison between the ANN-estimated and analytically obtained reliabilities is presented to highlight the estimation accuracy and convergence behavior.

6.6 Comparison $P_{up}(t)$ Vs $R(t)$ with Respect to Number of Epochs

Figure 6 presents the training progression of the two ANN models developed to estimate the system reliability and availability at the initial time point $t = 0$, using analytically computed reference values for comparison. Both models share a consistent architecture: a single-layer feedforward ANN with five hidden neurons using a logarithmic sigmoid (logsig) activation function. Training was conducted over 200 epochs in MATLAB software under identical hyperparameter conditions to ensure a uniform evaluation framework.

The blue curve illustrates the predicted values of reliability $R(0)$ across the training epochs. Initially, the reliability model starts with low output values, but quickly learns the underlying mapping, achieving close proximity to the actual reliability value of 1.0000 within the first 50 epochs. It stabilized at approximately 0.9849 in the remaining training iterations, indicating that the model converged early and remained consistent.

In contrast, the green curve, representing the availability model's output $A(0)$, starts at a higher initial value near 0.7840, which reflects the influence of repair and uptime-related factors inherent in availability calculations. Its learning curve shows a more gradual rise, with the model settling near 0.9797 by the end of the training. However, unlike the reliability curve, the convergence of the availability model slowed considerably after 100 epochs, suggesting a plateau in learning.

This comparative analysis reveals that, even with identical training conditions, the ANN with modeling reliability converged faster and exhibited greater early-stage learning efficiency. This is likely due to the relatively simpler structure of the reliability function, which does not account for the repair cycles or reactivation transitions. However, availability modeling, which captures more dynamic aspects of system operation, appears to require either extended training or more complex architectures for sharper convergence.

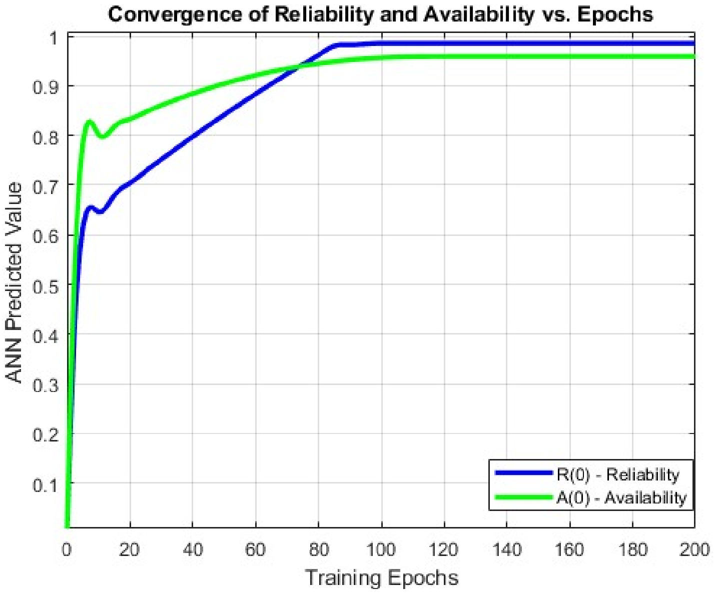

Figure 6. Convergence of reliability and availability Vs epochs.

The main objective of this study is not to claim superiority over all existing reliability models but to provide a consistent workflow for analyzing a warm standby repairable k -out-of- n network using a Markov/supplementary-variable formulation and demonstrate that a trained ANN can serve as a fast surrogate estimator of analytically obtained reliability and availability measures. Markov/CTMC-based techniques are widely used in repairable system reliability analysis, whereas neural networks have been employed as approximation tools for complex functional relationships. In this context, the comparison emphasized in this work is between the analytical model and the ANN estimator, where close agreement verifies that the ANN reproduces the analytical reliability/availability characteristics with low error and reduced computational burden for repeated evaluations within the studied parameter range.

Regarding generalizability, the proposed analytical–ANN workflow can be extended to other k-out-of-n network sizes by updating the corresponding state space and transition structure in the Markov/supplementary-variable formulation. In addition, different standby switching assumptions (e.g., instantaneous warm-standby switching versus delayed switching) can be incorporated by introducing appropriate transition rates or intermediate states. Likewise, alternative repair priority policies (e.g., prioritizing failed active units over standby units, or FCFS versus priority repair) can be modeled by modifying the repair-transition rules and their rates. Therefore, while the ANN acts as a surrogate estimator of the analytical measures, the underlying transition model can be adapted to study the sensitivity of reliability/availability to changes in network size, switching policy, and repair priorities.

The analytical results show a clear time-dependent degradation in system performance: both availability and reliability decrease as time increases for the warm standby repairable k-out-of-n laboratory network under the adopted failure–repair parameters (**Tables 2** to **5**). The ANN-based estimates tracked the analytical values closely at the same time points (**Tables 3** and **5**), which supports the use of the trained ANN as an accurate approximation of the analytical reliability/availability curves. These results also indicate that ANN can be used as a computationally efficient estimator for repeated evaluations within the studied parameter range.

The overall findings from the analytical and ANN-based estimation studies are summarized in the concluding section.

7. Conclusion

In this study, the reliability and availability of a warm standby repairable k-out-of-n computer laboratory network were evaluated using a continuous-time Markov framework and supplementary variable technique, and the resulting expressions were computed using MAPLE software. The analytical results show a monotonic decrease in both availability and reliability over time under the assumed failure–repair rates.

To support a fast evaluation, a feedforward ANN model was implemented in MATLAB software and trained using analytically generated reliability and availability samples. A single hidden layer architecture with logsig activation was adopted, and the ANN outputs were constrained within $[0, 1]$ while satisfying the initial conditions $P_{up}(0) = 1$and $R(0) = 1$. The close agreement between the ANN estimates and analytical values confirms that the ANN provides an accurate approximation of analytical reliability/availability characteristics. Therefore, the ANN component can be used as a computationally efficient estimator to reproduce the analytical curves and facilitate rapid evaluation under different parameter settings within the studied range.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This research project was funded by the University of Technology and Applied Sciences, Ibri, Oman through the Internal Research Funding Program, grant No IRG-IBRI-25-43. The author would like to thank the editor and anonymous reviewers for their comments that helped improve the quality of this work.

AI Disclosure

The authors declare that no assistance was taken by generative AI for writing this article.

References

Chand, G., & Singh, S.B. (2026). Availability modeling of three-state k-out-of-n system with delay time analysis incorporating hybrid hazard rate and copula. *Applied Mathematical Modelling*, *149*, 116282. https://doi.org/10.1016/j.apm.2025.116282.

Chantola, N., Singh, S.B., & Ekata. (2020). Reliability improvement of transformer using neural network approach. *International Journal of Reliability, Quality and Safety Engineering*, *27*(02), 2040005. https://doi.org/10.1142/S0218539320400057.

Chau, K.W. (2007). Reliability and performance-based design by artificial neural network. *Advances in Engineering Software*, *38*(3), 145-149. https://doi.org/10.1016/j.advengsoft.2006.09.008.

Ding, Z., Xu, T., Ye, T., & Zhou, Y. (2016). Online prediction and improvement of reliability for service oriented systems. *IEEE Transactions on Reliability*, *65*(3), 1133-1148. https://doi.org/10.1109/TR.2015.2504720.

Gebre, B.A., & Ramirez-Marquez, J.E. (2007). Element substitution algorithm for general two-terminal network reliability analyses. *IEEE Transactions on Reliability*, *39*(3), 265-275. https://doi.org/10.1080/07408170600735579.

Gupta, R., Ekata, & Batra, C.M. (2022). Performance assessment of solar transformer consumption system using neural network approach. *Baghdad Science Journal*, *19*(4), 865-874. https://doi.org/10.21123/bsj.2022.19.4.0865.

Jadhav, S., & Kumar, A. (2025). Feeding system's sensitivity and reliability analysis through Markov decision process. *International Journal of Mathematical, Engineering and Management Sciences*, *10*(2), 567-582. https://doi.org/10.33889/IJMEMS.2025.10.2.029.

Karunanithi, N., Whitley, D., & Malaiya, Y.K. (1992). Using neural networks in reliability prediction. *IEEE Software*, *9*(4), 53-59. https://doi.org/10.1109/52.143107.

Kiu, S.-W., & McAllister, D.F. (1988). Reliability optimization of computer-communication networks. *IEEE Transactions on Reliability*, *37*(5), 475-483. https://doi.org/10.1109/24.9866.

Koide, T., Shinmori, S., & Ishii, H. (2005). Efficient computation of network reliability importance on k-terminal reliability. *International Journal of Reliability, Quality and Safety Engineering*, *12*(3), 213-226. https://doi.org/10.1142/S0218539305001793.

Kumar, P., Kumar, A., & Chhetri, G. (2024). Reliability centered modeling of an automated waste sorting robotic arm system. *International Journal of Mathematical, Engineering and Management Sciences*, *9*(5), 1210-1225. https://doi.org/10.33889/IJMEMS.2024.9.5.064.

Lin, Y.K. (2007). Reliability evaluation for an information network with node failure under cost constraint. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *37*(2), 180-188. https://doi.org/10.1109 /TSMCA.2006.889478.

Lin, Y.-K. (2012). Evaluation of network reliability for a computer network subject to a budget constraint. *International Journal of Innovative Computing, Information and Control, 8*(6), 4205-4217.

Mahmood, M.A., Seah, W.K.G., & Welch, I. (2015). Reliability in wireless sensor networks: a survey and challenges ahead. *Computer Networks*, *79*, 166-187. https://doi.org/10.1016/j.comnet. 2014.12.016.

Munjal, A., & Singh, S.B. (2014). Reliability analysis of a complex repairable system composed of a 2-out-of-3: G subsystem and a series subsystem connected in parallel. *Journal of Reliability and Statistics Studies*, *7*(S), 89-111. https://journals.riverpublishers.com/index.php/JRSS/article/view/21635.

Nakagawa, T. (2005). *Maintenance theory of reliability*. Springer, London. ISBN: 978-1-85233-939-5. https://doi.org/10.1007/1-84628-221-7.

Nehra, V., Poonia, P.K., & Sirohi, A. (2023). Performance analysis of precision agriculture model using copula distribution and common cause breakdown. In *2023 13th International Conference on Cloud Computing, Data Science & Engineering* (pp. 388-395). IEEE. Noida, India. https://doi.org/10.1109/Confluence56041.2023.10048864.

Nelsen, R.B. (2006). *An introduction to copulas*. Springer, New York. ISBN: 978-0-387-28659-4. https://doi.org/10.1007/0-387-28678-0.

Poonia, P.K. (2021). Performance assessment of a multi-state computer network system in series configuration using copula repair. *International Journal of Reliability and Safety*, *15*(1-2), 68-88. https://doi.org/10.1504/IJRS.2021.119645.

Poonia, P.K. (2022). Performance assessment and sensitivity analysis of a computer lab network through copula repair with catastrophic failure. *Journal of Reliability and Statistical Studies*, *15*(1), 105-128. https://doi.org/10.13052/jrss0974-8024.1515.

Poonia, P.K. (2024). Exact reliability formula for precision agriculture through copula repair approach. *International Journal of System Assurance Engineering and Management*, *15*(8), 3725-3736. https://doi.org/10.1007/s13198-024-02372-1.

Poonia, P.K., & Tyagi, M. (2026). Reliability analysis of a series–parallel system under different failures using Gumbel-Hougaard copula repair. *Life Cycle Reliability and Safety Engineering*, *15*(1), 217-227. https://doi.org/10.1007/s41872-025-00346-1.

Raghav, D., Poonia, P.K., Gahlot, M., Singh, V.V., Ayagi, H.I., & Abdullahi, A.H. (2020). Probabilistic analysis of a system consisting of two subsystems in the series configuration under copula repair approach. *Theoretical Mathematics and Pedagogical Mathematics*, *27*(3), 137-155. https://doi.org/ 10.7468/jksmeb.2020.27.3.137.

Ram, M., Goyal, N., & Shivani (2025b). Standby system reliability modeling with cost and sensitivity analysis. *International Journal of Reliability, Quality and Safety Engineering*, *32*(2), 2450030. https://doi.org/10.1142/S021853932450030X.

Ram, M., Kharola, S., & Goyal, N. (2025a). Reliability and sensitivity analysis of a maintainable energy system under priority repair. *OPSEARCH*, *62*(4), 2047-2075. https://doi.org/10.1007/s12597-024-00868-9.

Ram, M., Singh, S.B., & Singh, V.V. (2013). Stochastic analysis of a standby system with waiting repair strategy. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *43*(3), 698-707. https://doi.org/10.1109/TSMCA.2012.2217320.

Saadon, S., Rambely, A.S., & Suradi, N.R.M. (2011). The role of computer labs in teaching and learning process in the field of mathematical sciences. *Procedia-Social and Behavioral Sciences*, *18*, 348-352. https://doi.org/10.1016/j.sbspro.2011.05.049.

Sharafat, A.R., & Ma'rouzi, O.R. (2009). All-terminal network reliability using recursive truncation algorithm. *IEEE Transactions on Reliability*, *58*(2), 338-347. https://doi.org/10.1109/TR.2009.2020120.

Sharma, S., Kumar, V., & Singh, S.B. (2025). Reliability measures of repairable k-out-of-n system using weighted exponential-Lindley distribution. *International Journal of Mathematics in Operational Research, 30*(2), 179-198. https://doi.org/10.1504/IJMOR.2025.148877.

Singh, V.V., Poonia, P.K., & Rawal, D.K. (2021). Reliability analysis of repairable network system of three computer labs connected with a server under 2-out-of-3 G configuration. *Life Cycle Reliability and Safety Engineering*, *10*(1), 19-29. https://doi.org/10.1007/s41872-020-00129-w.

Tamegai, N. (1980). Availability of Two k-server k-out-of-n: F Systems. *IEEE Transactions on Reliability*, *R-29*(1), 90-91. https://doi.org/10.1109/TR.1980.5220729.

Yeh, F.-M., Lu, S.-K., & Kuo, S.-Y. (2002). OBDD-based evaluation of k-terminal network reliability. *IEEE Transactions on Reliability*, *51*(4), 443-451. https://doi.org/10.1109/TR.2002.804736.

Zhang, H., Huang N., & Liu, H. (2014). Network performance reliability evaluation based on network reduction. *Reliability and Maintainability Symposium* (pp. 1-6). IEEE. Colorado Springs. https://doi.org/10.1109/RAMS.2014.6798440.

Publisher's Note- Ram Arti Publishers remains neutral regarding jurisdictional claims in published maps and institutional affiliations.