Overview

Modeling and simulation in engineering

Mehmet Sahinoglu*

This review article will explore the innovative and popular theme of engineering modeling and simulation, predominantly in the manufacturing industry and cybersecurity world, citing severe challenges, advantages and time- and budget saving solutions and its future. The power of simulation is not an exaggeration but an understatement. The favorable outcomes since the advent of digital computers and software revolution could not have been achieved, especially without the multiple benefits of statistical simulation, which underlies the widespread use of modeling and simulation in engineering and sciences, stretching from A (Astronomy) to Z (Zoology). This refers not only to research findings in verifying a certain piece of theory, such as that of the recently discovered Higgs Boson, but in testing new products to innovate new discoveries so as to make our universe a more peaceful place by modeling and simulating the future projects and taking precautions before disasters occur. The review explores a cross section of engineering modeling and simulation practices illustrating a window of numerical examples.

How to cite this article:
WIREs Comput Stat 2013, 5:239–266. doi: 10.1002/wics.1254

Keywords: discrete event/Monte Carlo; modeling; production; cyber-security; Bayesian; multistate

INTRODUCTION AND BRIEF HISTORY TO SIMULATION AND MOTIVATION

Computer modeling and simulation (M&S), as programs or networks of computers mimicking the execution of an abstract model of many natural systems from physical and life sciences to social and managerial sciences, and primarily engineering, have become an integral part of digital experimentation. M&S proves useful to estimate the performance of complex engineering systems when too prohibitive for analytical solutions. A simulation is defined as the reproduction of an event with the use of scientific models. A model is a physical, mathematical, or other logical representation of a system, process, or phenomenon. Time-independent static Monte Carlo (MC) or conversely dynamic Discrete Event Simulation (DES) to manage events in real time for engineering applications will be reviewed. Taxonomy-wise, simulated computer models may be stochastic or deterministic, and dynamic or static, and discrete or continuous.

Modern computer simulation developed in parallel with the rapid-growth of computer use during the development of the Manhattan Project in WWII to nondestructively model and simulate the nuclear detonation before it was destructively dropped on Hiroshima and Nagasaki in Japan in 1945. Therefore, the history of simulation is interesting and intriguing. Some earliest pioneers can be observed in Ref. 1 Lord Rayleigh in 1899 showed that a one-dimensional random walk without absorbing barriers could provide an approximate solution to a parabolic differential equation. In 1908 W.S. Gosset (with a nickname, Student) used experimental sampling to

Additional Supporting Information may be found in the online version of this article.

*Correspondence to: msahinog@aum.edu

Informatics Institute, Cybersystems and Information Security, Auburn University at Montgomery, Montgomery, AL, USA

Conflict of interest: The author has declared no conflicts of interest for this article.

help him towards his discovery of the distribution of the correlation coefficient and to bolster his faith in his so-called *t*-distribution.[2] A.N. Kolmogorov in 1931 showed the relationship between Markov stochastic processes and certain integro-differential equations. Stanislaw Ulam at Los Alamos labs performed simulation in 1945 during the WWII in the bomb-building Manhattan Project before proposing the Teller-Ulam thermonuclear weapon design. Ulam suggested first the 'Russian Roulette' and 'splitting' methods, for evaluating complicated mathematical integrals for nuclear chain reactions that later led to the systematic Monte Carlo methods by von Neumann, Metropolis and others. John von Neumann explored the behavior of neutron chain reactions in fission devices using statistical sampling methods in 1948 (such as the acceptance–rejection method) employing the newly developed electronic computing techniques. Neumann proposed the agent-based Von Neumann Machine,[3] a theoretical machine capable of reproduction following detailed instructions to copy itself. Ulam suggested a machine as a collection of cells on a grid. The idea intrigued von Neumann, who created the first of the devices later termed, cellular automaton.[4] John Conway constructed the well-known Game of Life,[5] operated in a virtual world in the form of a two-dimensional checkerboard. A team headed by N. Metropolis using the ENIAC Computer in 1948 carried out what's contemporarily known as modern Monte Carlo calculations.

Computer simulation has been widely used in engineering systems to validate the effectiveness of tentative decisions regarding a new plan or schedule, or its outcomes, without actually experiencing the actual conditions, which could in actuality cost more resources or partial to full destruction such as in the simulation of the nuclear bomb. In a book titled, *Simulation Engineering*, by Jim Ledin in 2001,[6] the author outlines his twofold purpose as follows: (1) simulation engineering (SE) is the application of engineering discipline to the development of good simulations. (2) Similarly, SE occurs when simulations become part of an engineering process when applied as tools to develop better products and test processes with a greater efficiency for different types of complex embedded systems. The latter purpose (2) is the subject matter of this review article. The IEEE June 2012 Spectrum issue, emphasized that the Modeling and Simulation effect is a creative and time-saving topic of interest ranging from automotive engineering of hybrid vehicles to finding solutions to treating nuclear waste, and upgrading the nuts and bolts of the Electrical Power (Smart) Grid and moreover, supercomputing research.[7]

GENERIC THEORY—CASE STUDIES ON GOODNESS OF FIT FOR UNIFORM NUMBERS

A formal scientific theory of simulation, to verify a validated model so as to mimic a physical or a social system, does not exist in terms of conventional math-statistical theorems and their subsequent proofs. However, heuristic modeling formalisms at an advanced level for engineers through cellular automaton for Monte Carlo and Discrete Event Simulations are studied by Zeigler et al.[8] (Ch. 4), although these formalisms do not lend themselves to easy algorithmic implementations for practicing engineers or scientists as this review article purports to. Moreover, the fundamental process of verifying sequences of uniform deviates from an associated generator where Ho: Uniform Random Sequence is (quasi) random versus Ha: Sequence is not random, is an accepted technique. For instance, χ^2 tests, such as those by Leven and Wolfowitz[9] and Knuth,[10] are popularly well-accepted math-statistical scientific practices to theorize the verifiability of uniform random numbers essential to the realm of statistical simulation. In order to clarify the validation of the above stated Ho: Random Sequence versus Ha: Not Random Sequence, the commercial JAVA code's uniform number generator will be tested for randomness, as illustrated in a series of screenshots from Figures A1–A9 in Appendix A by using Stewart's JAVA program to implement Knuth's technique.[11] The results show that by law of large numbers for only $n \geq N \approx 50{,}000$; $E(\theta) \to 0.5$ with probability 1, for $\theta \sim$ Uniform(0, 1) from the uniform number generator imbedded in the Java code, 'Ho: Random Sequence' is not rejected. Therefore $n = 50{,}000$ runs is a new standard for attaining quasirandomness; not 5000 anymore as practiced in 1980s.

WHY CRUCIAL TO ENGINEERING—MANUFACTURING AND CYBER DEFENSE ISSUES

The power of simulation is prevalent as the audio-visual Ref 12 favorably explains certain topics related to production and manufacturing engineering. In 'Modeling and Simulation in Manufacturing and Defense Systems Acquisition', the Board on Manufacturing and Engineering Design (BMED) emphasized the importance of modeling and simulation in not only making the right decisions but also incurring fewer expenses.[13] Similarly, the Wychavon (UK) council has adopted manufacturing industry's simulation model to reduce waste and improve performance.[14] Since the US manufacturing industry is challenged by increased global competition and price erosion, one can benefit from manufacturing simulation to eliminate bottlenecks, enhance lean manufacturing, optimize capacity planning and optimize production output. In an Annotated Discrete Event Simulation Bibliography, there exist 325 articles on manufacturing simulation as cited in Ref 15. A certain bibliography displays 112 publications on '*Load Models for Power Flow and Dynamic Performance Simulation*' by the IEEE Transactions on Electric Power Systems.[16]

On the contrary, there are fewer simulation studies in cybersecurity-and-defense-related theoretical and applied research. In their 2007 article as titled, '*Cyber Attack Modeling and Simulation for Network Security Analysis*', the authors Kuhl et al. discuss a simulation modeling approach to represent computer networks and intrusion detection systems (IDS) to efficiently simulate cyber-attack scenarios in order to test and evaluate cyber security systems.[17] You-Tube-based audio-visual Cybersecurity Simulation roundtable[18] underlines the power of simulation in cybersecurity scenarios. Under 'War game reveals US lacks cyber-crisis skills' in a war game,[19] sponsored by a nonprofit group and attended by former top-ranking national security officials, laid bare that the US government lacks answers to such key questions. Former Clinton press secretary Joe Lockhart said that people would be scared by the simulation but he added, '...that's a good thing.' Sahinoglu in his 2007 Wiley textbook, *Trustworthy Computing: Analytical and Quantitative Engineering Evaluation*,[20] considers modeling and simulation of individual components and systems toward assessment of security risk, in addition to his publications where theoretical models are confirmed using Monte Carlo and Discrete event simulation runs.[21–24] Further, certain manufacturing- and cybersecurity-themed examples will be reviewed through working details of how the modeling should be validating the physical model and the subsequent simulation computationally verifying the solutions accurately and cost effectively. These reviews are the tips of the iceberg, as industries will continue to design and discover new products and services by M&S.

Figure 1 displays the interaction between the process of building a model by focusing on the interplay between (1) experimental results, (2) simulation results, and (3) theoretical predictions as displayed in Ref 25. A favorable example of this interplay is presented in a recent *WIREs* article titled *Cloud Computing*[26] (Figure 8, p. 55), which displays an *experimental* scenario for a trivial Cyber Cloud. On the other hand Ref 26 (Figure 9, p. 57) outlines the Markovian *theoretical* predictions followed by the *simulation* results for the same scenario of two 1-GB units serving a constant load of 1.5 GB for 13 cycles. The resulting availability of this small Cloud: (1) 0.307 for *Experimental*, (2) 0.305 for *Simulation* after 1 million runs, or trials and (3) 0.331 for Markov *Theoretical*, allowing a negligible error content, which diminishes to less than 3% as the size of the experiment increases from a few to many hundreds of units. In the event of large cyber CLOUDS such as those with 398 units, the authors showed that the experimental approach was infeasible, and theoretical result was not mathematically tractable. However, supercomputer-driven programming worked for days regarding the basic two-state assumptions, crunching 2^{398} ($\gg 10^{100}$) Markov states to 93.8% reliability. DES result was a satisfactorily comparable 90.5%.

A CROSS SECTION OF MODELING AND SIMULATION ISSUES IN MANUFACTURING

Simulation use in production is not new. For the sake of a few examples, various authors from 25 years ago published articles on simulating flexible manufacturing systems (FMS), machine utilizations and production rates, and modeling of Automated Manufacturing Systems (AMS).[27–29] Given the advances in pervasive computing regarding communication networks, as well as recently popularized large scale cloud computing in cyber networks; quantitative risk assessment of a manufacturing unit and their network availability have become challenging tasks. An often overlooked fact is that many real-life grid units such as routers or servers in cyber physical systems to the manufacturing assemblies in automotive or avionics, etc. and the intricate telephony networks (wired or wireless), and water-supply networks or hydroelectric dams, do not operate in an idealized simple setting of either full or zero capacity. This fact therefore necessitates the inclusion of degrees of derated (in-between *UP* and *DOWN* states) capacity. Because of lack of closed-form solutions in the three-state model including *DERATED* as opposed to that of the conventional *UP*-and-*DOWN* dichotomous two-state model, a summary of three or multistate system inferential analysis will be reviewed by using Monte Carlo simulations. This process will employ the empirical Bayesian principles to estimate the full and derated availability probability distributions. The historical failure and repair data, or operating (full or derated) and nonoperating hours, as the input data, will

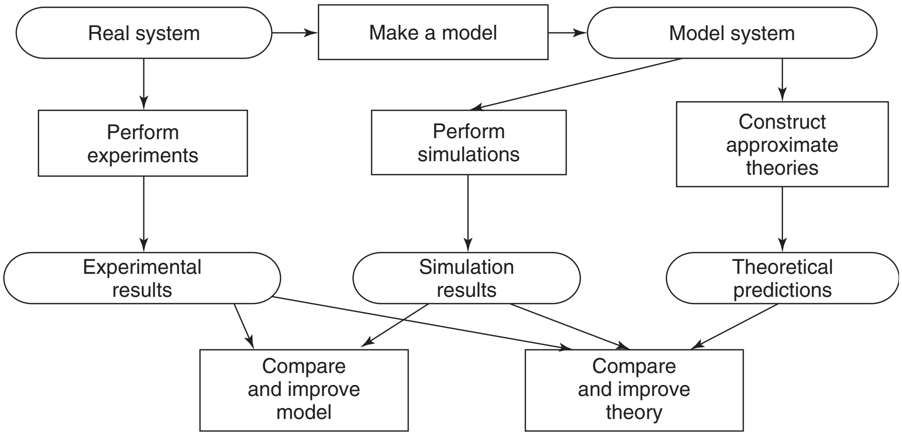

FIGURE 1 | Computer modeling and interplay between experiments, simulation, and theory.[25]

be used along with prior parameters for an empirical Bayesian analysis. The results satisfactorily lend themselves to statistical inference for multiple states other than the traditional binary assumption (*UP* or *DOWN*), an outcome which can prove very useful to the manufacturing industry. In the past, various articles have studied a similar problem. For instance, '*A Hybrid Markov system dynamics approach for availability analysis of degraded systems*' by Rao et al[30]; similarly, Lins and Droguett study '*Multi-objective Optimization of Availability and Cost in Repairable Systems Design Via Genetic Algorithms and Discrete Event Simulation*.[31] '*Reliability and Availability Analysis of Three-state Device Redundant Systems with Human Errors and Common-cause Failures*', by Shah and Dhillon studies somewhat similar but still different topics.[32] The primary difference between the above listed three references and this review article is the empirical Bayesian treatment of the three states to estimate their probability distributions by Monte Carlo simulations based on the Sahinoglu–Libby probability density function, originally derived independently by both Sahinoglu and Libby in 1981.[33–37] The closest among these three articles, that is, by Rao et al.[30] uses only four transition rates in a three-state Markov model whereas Sahinoglu's model uses all six transitions.[38,39] However, this review article's simulation approach is even more powerful and flexible as the application can be extrapolated based on identical principles to four or more states, whereas reference by Rao et al.[30] deals solely with differential equations limited in scope. Others, Lins et al.[31] and Shah et al.,[32] are on slightly different but not identical topics; all of which do not employ modeling and simulation techniques or generate closed form statistical probability density function (p.d.f.) expressions and derivations.

Modeling and Simulation of Multistate Production Units and Systems in Manufacturing

Most research articles or books on reliability theory are devoted to traditional binary reliability models allowing for only two possible states for a system and its components: perfect functionality or complete failure. However, many real-world systems are composed of multistate components which have different performance levels and several failure modes with varying effects on the entire system performance. Such systems are called multistate systems (MSS).[40] Examples of MSS are cyber systems where the unit performance is characterized by the data processing speed or server gigabyte capacity and similar to electric power systems, where the generating unit performance is depicted by its generating capacity. In the electric power supply system of generating facilities, each generator can function at different levels of capacity with a given probability. This may result from the outages of several auxiliaries such as pulverizers, water pumps, fans, boilers, etc. Billinton and Allan[41] describe a three state 50 MW generating unit. The performance rates (generating capacity) corresponding to these three states and probabilities of the three states which sum to unity are presented as follows: Probability of State 1 (50 MW capacity) = 0.960, Probability of State 2 (30 MW capacity) = 0.033 and Probability of State 3 (0 MW capacity) = 0.007. Therefore, the reliability analysis of MSS is much more complex compared to binary-state systems. From the mid-1970s until now, various books and research articles focusing on MSS reliability were published.[20,40–45] However, these works are deterministic, and not probabilistic, thus not lending themselves to probability distribution functions other than a single summary measure.

Therefore statistical inference cannot be conducted. This article reviews methodology for the estimation of the probability distributions of three-state (now including a new derated or degraded state beyond the binary assumption of *UP* or *DOWN*) repairable hardware units or components by using Monte Carlo simulations in employing the statistical random number generation techniques.

The power of simulation once again flexes its muscle as a favorable exit out of this theoretical impasse. The Monte Carlo technique remains the only available feasible way to solve the proposed three-state problem, whose math-statistical closed-form solution does not actually exist. This is mainly because the three-state Markov model's random variables' (*UP*, *DOWN*, *DER*) probability distributions cannot be derived through math-statistical transformations due to mathematical intractability and lack of sufficient statistical theory. The probability density function of the Forced Outage Rate (FOR) was earlier analyzed in a textbook by the primary author, who designated that the Sahinoglu–Libby (SL) probability model can be used if certain underlying assumptions hold.[20,33–35] Libby and Novick independently have studied multivariate generalized beta distributions for utility assessment; however their analysis was for only two-states similarly, not for multistate hence the term, G3B: Generalized 3-Parameter Beta.[36,37] The failure and repair rates were taken to be the generalized gamma random deviates where the corresponding gamma shape and scale parameters, respectively, were not equal. The two-state SL density was shown to default to that of a standard two-parameter beta density function when the shape parameters are identical. The stochastic method proposed was superior to estimating availability by dividing total uptime by exposure time. Examples had shown the validity of this method to avoid over- or under-estimation of availability when only small samples or insufficient data exist for the historical life cycles of units. In this article, however, additionally we shall also review, similar to the two-state SL, a computational three-state simulation model. Because of the infeasibility of closed-form solutions, the analysis will be carried out using Monte Carlo simulations, obeying the Bayesian principles similar to Chapter 5 of the author's textbook.[20] In studying large capacity production units, it is necessary to consider the probabilities associated with one or more forced derated states rather than accepting the unit being either available or unavailable, according to Billinton.[44] Following the Monte Carlo simulations, analytical p.d.f.s of the multiple states will be approximated using their associated moments.

Two-State Sahinoglu–Libby Probability Model of Production Units (Closed-Form Solution)

In using the distribution function technique, the p.d.f. of $FOR = q = \lambda/(\lambda+\mu)$ is obtained first by deriving its cumulative density function (c.d.f.), i.e. $G_Q(q) = P(Q \le q) = P(\lambda/(\lambda+\mu) \le q)$. Then, taking its derivative to obtain $g_Q(q)$ as per Eqs (5A.1)–(5A.18) in Appendix 5A, on pp. 26–32 of Ref 20 and Ref 33, p. 1487, and also in Ref 34; $g_Q(q)$ is as follows:

$$g_Q(q) = \frac{\Gamma(a+b+c+d)}{\Gamma(a+c)\Gamma(b+d)} \times \frac{(\xi+x_T)^{a+c}(\eta+y_T)^{b+d}(1-q)^{b+d-1}q^{a+c-1}}{[\eta+y_T+q(\xi+x_T-\eta-y_T)]^{a+b+c+d}}$$

(skipping steps)

$$\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}(1-q)^{\beta-1}q^{\alpha-1}\left[\frac{1}{1+q(L-1)}\right]^{\alpha+\beta} L^{\alpha}. \tag{1}$$

Note that $g_Q(q)$ is the p.d.f. of the random variable $Q = FOR$, where $\alpha = a+c$, $\beta = b+d$, $\beta_1 = \xi + x_T$, and $\beta_2 = \eta + y_T$; and $0 \le q \le 1$. If $L = (\beta_1/\beta_2)$ for SL (α, β, L) or $\beta_1 = \beta_2$, the usual two-parameter beta p.d.f. is obtained. An alternative original derivation of the same p.d.f. termed under generalized multivariate beta distribution is given by Libby in 1981 and 1982.[36,37] The expression in Eq. (1) can also be reformulated in terms of $SL(\alpha = a+c$, $\beta = b+d$, $L = (\beta_1/\beta_2))$, as follows:

$$g_Q(q) = \frac{L^{\alpha+c}q^{\alpha+c-1}(1-q)^{b+d-1}}{B(b+d,a+c)[1-(1-L)q]^{a+b+c+d}}, \tag{2}$$

where

$$B(b+d,a+c) = \frac{\Gamma(a+c)\Gamma(b+d)}{\Gamma(a+b+c+d)}, \text{ and } L = \frac{\xi+x_T}{\eta+y_T}. \tag{3}$$

Note if $L = 1$, Sahinoglu-Libby p.d.f. reduces to a standard Beta (α, β) p.d.f. See Figure 2 for 'r = availability' and 'q = unavailability' confidence plots where $r = 1 - q$. Densities of SL (or $G3B$) distributions have been cited in Refs 33–37 for a variety of L values. From a strictly mathematical point of view, the presence of the parameter L allows the SL p.d.f. to take a variety of shapes besides the standard Beta(α, β) where $L = 1$. For example, when $\alpha = \beta$, the standard Beta(α, α) is symmetric with a mean at 0.5. However, the SL (α, α, L)

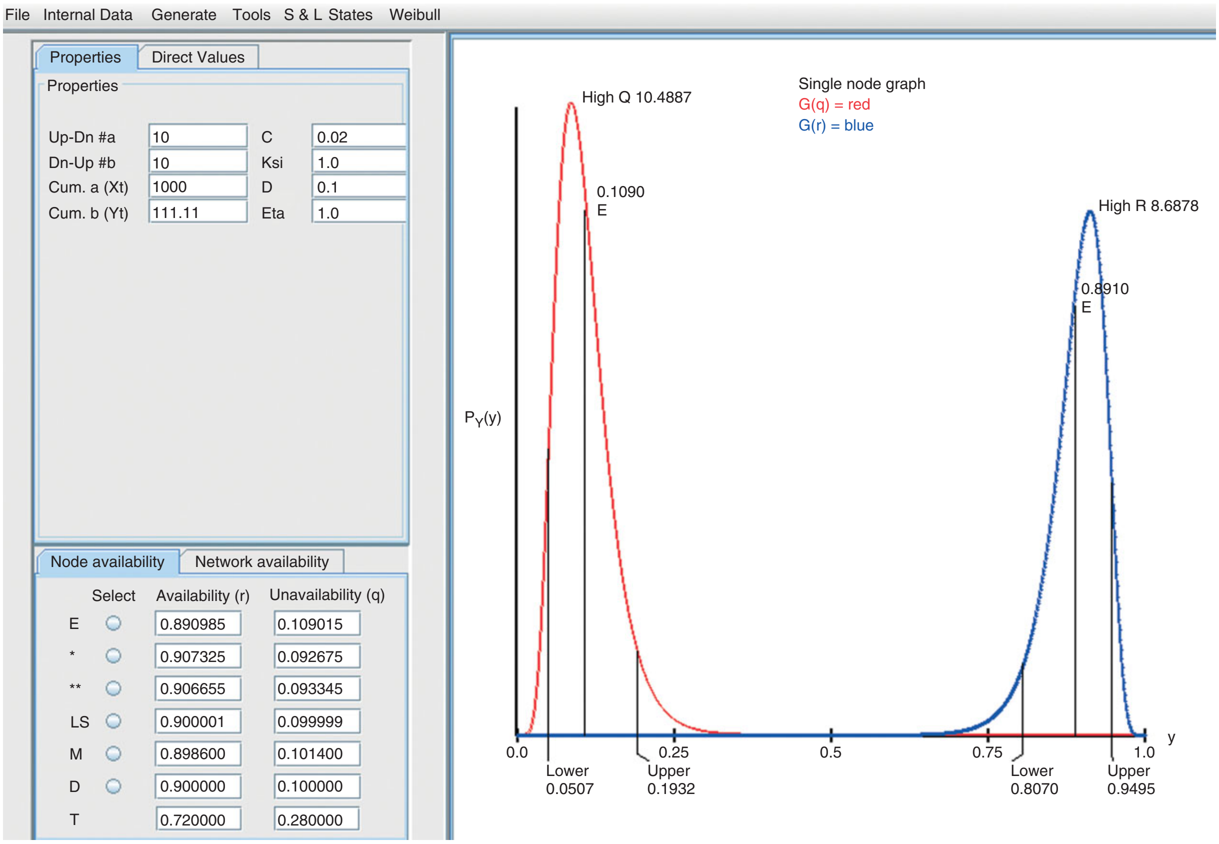

FIGURE 2 | Given the input table, the p.d.f. of the two-state *SL* is plotted for *UP* (*r*) and *DOWN* (*q*) for 90% confidence analytically showing mode (*m*), mean (*E*) with upper & lower confidence bounds.

distribution is not necessarily so, and can be skewed positively or negatively, depending on $L > 1$ and $L < 1$ respectively, because the mode, skewness, and kurtosis of SL random variable now also depend on L. For $0 < L < 1$, the SL p.d.f. stays below the plot of the related standard Beta near zero but crosses the latter to become the greater of the two p.d.f.s at a point[36,37]:

$$y_0 = [1 - L^{\alpha_1/(\alpha_1+\alpha_2)}]^{-1} - (1 - L)^{-1}. \quad (4)$$

The reverse action holds true for $L > 1$ with the same crossing point, y_0. The major drawback to the distribution is that there is no closed form for finite estimates of the moments. The moment generating function for the univariate SL distribution is an infinite series.[36,37]

Three-State Sahinoglu Probability Model of Production Units (Monte Carlo Simulation)

In studying large capacity generation (power) or production (or cyber-physical) units, it may be necessary to consider the probabilities associated with one or more forced derated-outage states as in multistate, rather than considering the unit as being either available or unavailable.[40–42,44,45] In summary, there are gray areas or in-between capacities which are called derated or degraded states. However in this review article, we will only consider a single derated state rather than multiple ones, which may well exist in practice such as in 50%, 60%, or 75% derated capacity. But now, we have not only full-*FOR* but also derated-*FOR* (or *DFOR*), that will be equal to the total derated operating time over the total exposure time. That is, *DFOR* = *DER* time/(*UP* time + *DER* time + *DOWN* time). It is also well documented that any calculated *FOR* or *DFOR* is not only a constant but also a specific single realization of its random variable.[20] The probability density function of the *FOR* by empirical Bayesian analysis was identified in Section *Two-State Sahinoglu-Libby Probability Model of Production Units* (*Closed-Form Solution*) to be the Sahinoglu–Libby (*SL*) probability density, where certain underlying assumptions hold. However, we shall review above and beyond a traditional closed-form two-state *SL*; namely, a three-state *SL* where the transition rates are *gamma* distributed (see derivations in subsections of *Three-State Sahinoglu*

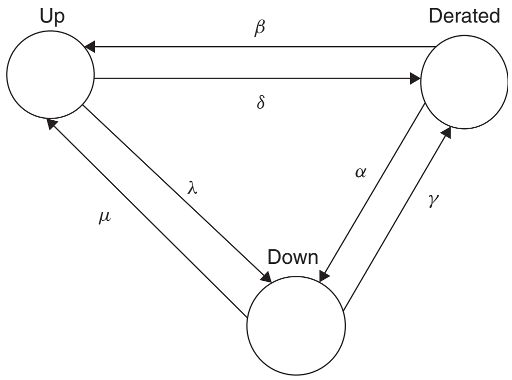

FIGURE 3 | Three-state Markov diagram of a repairable hardware unit with *UP*, *DOWN* and *DER* states.

Probability Model of Production Units (*Monte Carlo Simulation*)). Let us examine and review the following state space diagram in Figure 3 by Billinton from his textbook.[44]

Let $\lambda =$ transition rate from *UP* (fully operational) to *DOWN* (forced outage) state. Let $\mu =$ transition rate from *DOWN* to *UP* state; $\delta =$ transition rate from *UP* to *DER* (partially forced outage) state; $\beta =$ transition rate from *DER* (partially forced outage) to *UP* state; $\alpha =$ transition rate from *DER* to *DOWN* state; $\gamma =$ transition rate from *DOWN* to *DER* state. Using Figure 3 given, by changing to the Greek variables from the Latin originals (a–f) cited in the same reference[44] (p.156, Fig. 4.2); the time-dependent but steady state probabilities of occupying one of the three states are given as follow from (5)–(7), assuming negative exponential densities for each state's sojourn time, which will converge to:

$$P(UP) = FOR = \frac{\mu\beta + \mu\gamma + \alpha\beta}{DENOMINATOR}, \quad (5)$$

$$P(DERATED) = DFOR = \frac{\delta\mu + \delta\alpha + \lambda\mu}{DENOMINATOR}, \quad (6)$$

$$P(DOWN) = 1 - P(UP) - P(DERATED) = \frac{\lambda\beta + \lambda\gamma + \delta\gamma}{DENOMINATOR}, \quad (7)$$

where

$$DENOMINATOR = \mu\beta + \mu\gamma + \alpha\beta + \lambda\beta + \lambda\gamma + \delta\mu + \delta\alpha + \lambda\mu + \delta\gamma. \quad (8)$$

A closed form solution of the three-state SL is intractable and analytically impossible in this setting with six random variables, as compared solely to the two variables in Section *Two-State Sahinoglu-Libby Probability Model of Production Units* (*Closed-Form Solution*). We will therefore have to simulate the $P(UP)$, $P(DER)$ and $P(DOWN)$ from Eqs (5)–(7) by generating the recursive Monte Carlo simulated deviates of the state transition rates. Empirical Bayesian analysis will be pursued through deriving first the conditional posterior densities of the six transition rates from subsections of *Three-State Sahinoglu Probability Model of Production Units* (*Monte Carlo Simulation*), and using random uniforms for generating the transitions that constitute the probabilities in Eqs (5)–(7). See Figure 4 for a sample draft scenarios to illustrate transitions of Figure 3.

UP-to-DOWN Failure Transition Rate (λ), *for example, from* $x1$ *to* $w1$, *or* $x2$ *to* $w2$ *in Figure 4*

Let $a =$ number of occurrences of *UP* (operating) times before *DOWN* (recovery)

$X_i \sim \lambda e^{-\lambda X}$.

$x_T = \sum_{1}^{a} X_i =$ total *UP* (operating) times before going *DOWN* (recovery) for 'a' such occurrences.

$\lambda =$ full *UP*-to-*DOWN* rate.

$c =$ shape parameter of *gamma* prior for the full failure rate λ.

$\xi =$ inverse scale parameter of *gamma* prior for the full failure rate λ.

Now let the failure rate, λ have a *gamma* prior distribution:

$$\theta_1(\lambda) = \frac{\xi^c}{\Gamma(c)}\lambda^{c-1}\exp(-\lambda\xi), \lambda = 0. \quad (9)$$

The joint likelihood of the *UP*-time random variables is

$$f(x_1, x_2, \ldots, x_a|\lambda) = \exp(-xT\lambda), \quad (10)$$

the joint distribution of data and prior becomes:

$$k(\underline{x}, \lambda) = f(x_1, x_2, \ldots, x_a, \lambda) = \frac{\xi^c}{\Gamma(c)}\lambda^{a+c-1}\exp[-\lambda(x_T + \xi)]. \quad (11)$$

Thus, the posterior distribution for the random variable λ is

$$h_1(\lambda|\underline{x}) = \frac{\xi^c}{\Gamma(c)}\lambda^{a+c-1}\exp[-\lambda(x_T+\xi)] \div \frac{\xi^c}{\Gamma(c)}(x_T+\xi)^{-1}\Gamma(a+c) = \frac{1}{\Gamma(a+c)}(x_T+\xi)\lambda^{a+c-1}\exp[-\lambda(x_T+\xi)], \quad (12)$$

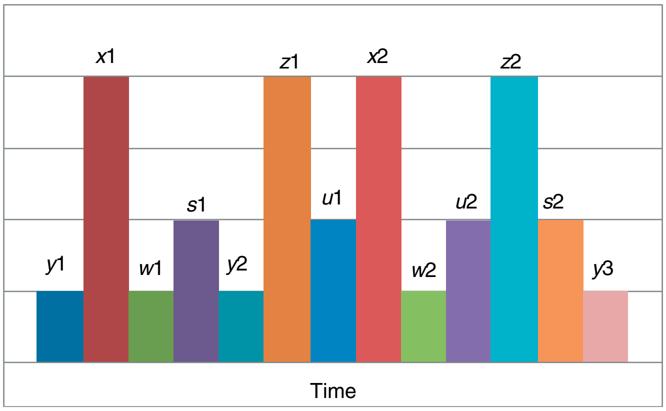

FIGURE 4 | A sample illustration of feasible transitions from Figure 3 implemented to subsections of *Three-State Sahinoglu Probability Model of Production Units* (*Monte Carlo Simulation*).

which is also distributed as $Gamma[a + c, (x_T + \xi)^{-1}]$. Note that $\underline{x}$ is a vector.

DOWN-to-UP Recovery Transition Rate (μ), for example, from y1 to x1, or y2 to z1 in Figure 4

Let b = number of occurrences of *DOWN* (recovery) times before *UP* (operating)

$Y_i \sim \mu e^{-\mu Y}$

$y_T = \sum_1^b Y_i$ = total *DOWN* (recovery) times before going *UP* for 'b' many such occurrences

μ = full recovery (*DOWN*-to-*UP*) rate

d = shape parameter of *gamma* prior for the full recovery rate μ

η = inverse scale parameter of gamma prior for the full recovery rate μ

Now let the full recovery rate, μ, have a *gamma* prior distribution:

$$\theta_2(\mu) = \frac{\eta^d}{\Gamma(d)}\mu^{d-1}\exp(-\mu\eta),\ \mu > 0. \qquad (13)$$

The joint likelihood of the *DOWN*-time random variables is

$$f(y_1, y_2, \ldots, y_a|\mu) = \mu^b \exp(-y_T\mu). \qquad (14)$$

The joint distribution of data and prior becomes:

$$k(\underline{y}, \mu) = f(y_1, y_2, \ldots, y_b, \mu)$$
$$= \frac{\eta^d}{\Gamma(d)}\mu^{b+d-1}\exp[-\mu(y_r + \eta)]. \qquad (15)$$

Thus, similarly skipping two intermediate steps, the posterior distribution for μ is

$$h_2(\mu|\underline{y}) = \frac{1}{\Gamma(b+d)}(y_T + \eta)\mu^{b+d-1}\exp[-\mu(y_T + \eta)], \qquad (16)$$

which is also distributed as $Gamma[b + d, (y_T + \eta)^{-1}]$. Note that $\underline{y}$ is a vector.

UP-to-DER Failure Transition Rate (δ), e.g. from z1 to u1, or z2 to s2 in Figure 4

Let o = number of occurrences of *UP* times before *DER*

$z_T = \sum_1^o Z_i$ = total *UP* times before going *DER* for 'o' many of such occurrences.

$Z_i \sim \delta e^{\delta Z}$.

δ = *UP*-to-*DER* failure rate.

e = shape parameter of gamma prior for the *UP*-to-*DER* failure rate δ.

Δ = inverse scale parameter of gamma prior for the *UP*-to-*DER* failure rate δ.

Now let the *UP*-to-*DER* failure rate δ have a *gamma* prior distribution:

$$\theta_3(\delta) = \frac{\Delta^e}{\Gamma(e)}\delta^{e-1}\exp(-\delta\Delta),\ \delta > 0. \qquad (17)$$

Thus, similarly skipping two intermediate steps, the conditional posterior density of δ becomes:

$$h_3(\delta|\underline{z}) = \frac{1}{\Gamma(o+e)}(z_T + \Delta)\delta^{o+e-1}\exp[-\delta(z_T + \Delta)], \qquad (18)$$

which is also distributed as $Gamma\ [o + e, (z_T + \Delta)^{-1}]$. Note that $\underline{z}$ is a vector.

DER-to-UP Recovery Transition Rate (β), for example, from u1 to x2, or u2 to z2 in Figure 4

Let k = number of occurrences of *DER* times before *UP*

$u_T = \sum_1^k U_i$ = total *DER* failure times before going *UP* for 'k' many of such occurrences.

$U_i \sim \beta e^{-\beta U}$.

β = *DER*-to-*UP* recovery rate.

ϕ = shape parameter of *gamma* prior for the *DER*-to-*UP* recovery rate β.

f = inverse scale parameter of gamma prior for the *DER*-to-*UP* recovery rate.

Now let the *DER*-to-*UP* recovery rate β have a *gamma* prior distribution:

$$\theta_4(\beta) = \frac{\phi^f}{\Gamma(f)}\beta^{f-1}\exp(-\beta\phi),\ \beta > 0. \tag{19}$$

Thus, similarly skipping two intermediate steps, the conditional posterior density of β:

$$h_4(\beta|\underline{u}) = \frac{1}{\Gamma(k+f)}(u_T+\phi)\beta^{k+f-1}\exp[-\beta(u_T+\phi)], \tag{20}$$

which is also distributed as *Gamma* $[k+f,\ (u_T+\phi)^{-1}]$. Note that $\underline{u}$ is a vector.

DER-to-DOWN Failure Transition Rate (α); for example, from s1 to y2, or s2 to y3 in Figure 4

Let j = number of occurrences of *DER* failure times before *DOWN*

$s_T = \sum_1^j S_i$ = total *DER* failure times before going *DOWN* for 'j' many such occurrences.

$S_i \sim \alpha e^{-\alpha S}$.

α = *DER*-to-*DOWN* failure rate.

g = shape parameter of *gamma* prior for *DER*-to-*DOWN* failure rate α.

ψ = inverse scale parameter of *gamma* prior for *DER*-to-*DOWN* failure rate α.

Now let the *DER*-to-*DOWN* failure rate α have a *gamma* prior distribution:

$$\theta_5(\alpha) = \frac{\psi^g}{\Gamma(g)}\alpha^{g-1}\exp(-\alpha\psi),\ \alpha > 0. \tag{21}$$

Thus, similarly skipping two intermediate steps, the conditional posterior density of α:

$$h_5(\alpha|\underline{s}) = \frac{1}{\Gamma(j+g)}(s_T+\psi)\alpha^{j+g-1}\exp[-\alpha(s_T+\psi)], \tag{22}$$

which is also a *Gamma* $[j+g,\ (s_T+\psi)^{-1}]$. Note that $\underline{s}$ is a vector.

DOWN-to-DER Recovery Transition Rate (γ), e.g. from w1 to s1, or w2 to u2 in Figure 4

Let p = number of occurrences of *DOWN* times before *DER*

$w_T = \sum_1^p W_i$ = total *DOWN* times before going *DER* for 'p' many such occurrences.

$W_i \sim \gamma e^{-\gamma W}$.

γ = *DOWN*-to-*DER* recovery rate.

h = shape parameter of *gamma* prior for the *DOWN*-to-*DER* recovery rate γ.

π = inverse scale parameter of *gamma* prior for the *DOWN*-to-*DER* recovery rate γ.

Now let the *DOWN*-to-*DER* recovery rate γ have a *gamma* prior distribution:

$$\theta_6(\gamma) = \frac{\pi^h}{\Gamma(h)}\gamma^{h-1}\exp(-\gamma\pi),\ \gamma > 0. \tag{23}$$

Thus, similarly skipping two intermediate steps, the conditional posterior density of γ:

$$h_6(\gamma|\underline{w}) = \frac{1}{\Gamma(p+h)}(w_T+\pi)\gamma^{p+h-1}\exp[-\gamma(w_T+\pi)], \tag{24}$$

which is also distributed as *Gamma* $[p+h,\ (w_T+\pi)^{-1}]$. Note that $\underline{w}$ is a vector.

Statistical Simulation of Three-State Units to Estimate the Density of *UP*, *DOWN*, and *DER*

Table 1 displays the input data as tabulated for the following example covering the first five episodes of six different sojourn times (see Figures 3 and 4).

The cumulative probabilities of states are calculated by Monte Carlo Simulation method using input from Table 1 as follows in Tables 2–4 for *UP*, *DER* and *DOWN* states in 100, 1000, and 10,000 simulation runs, respectively. Figures 5–7 using Eqs (5)–(7) will convert these tabulations into cumulative frequency plots utilizing the six transitions of Figure 3 in subsections of *Three-State*

TABLE 1 | An Input Data Example for the Monte Carlo Simulations of *UP*, *DOWN*, and *DER* States for the first 5 episodes (#Events)

#Events	Exposure Time	Shape Parameter	Scale Parameter	Transition Rate
$a = 5$	$X_T = 25$	$c = 0.2$	$\xi = 1$	λ
$b = 5$	$Y_T = 5$	$d = 2$	$\eta = 0.5$	μ
$o = 5$	$Z_T = 10$	$e = 1$	$\Delta = 0.5$	δ
$k = 5$	$U_T = 20$	$f = 0.5$	Ø = 1	β
$j = 5$	$S_T = 10$	$g = 1$	$\psi = 0.5$	α
$p = 5$	$W_T = 15$	$h = 2$	$\pi = 1$	γ

TABLE 2 | *UP* STATE EQ(5)

Cumulative Density	<0.1	<0.2	<0.3	<0.4	<0.5	<0.6	<0.7	<0.8
100 *simulation runs*								
Total count	0	2	19	62	92	100	100	100
Cumulative Probability	0	0.02	0.19	0.62	0.92	1	1	1
1000 *simulation runs*								
Total count	0	21	185	597	885	978	998	1000
Cumulative Probability	0	0.021	0.185	0.597	0.885	0.978	0.998	1
10,000 *simulation runs*								
Total count	0	187	2000	5874	8815	9816	9984	10,000
Cumulative Probability	0	0.0187	0.2	0.5874	0.885	0.978	0.998	1

TABLE 3 | *DERATED* STATE EQ(6)

Cumulative Density	<0.05	<0.1	<0.15	<0.2	<0.25	<0.3	<0.35	<0.4	<0.45
100 *simulation runs*									
Total count	0	10	43	80	97	100	100	100	100
Cumulative Probability	0	0.1	0.43	0.8	0.97	1	1	1	1
1000 *simulation runs*									
Total count	0.34	181	552	829	954	984	995	998	999
Cumulative Probability	0.034	0.181	0.552	0.829	0.954	0.984	0.995	0.998	0.999
10,000 *simulation runs*									
Total count	34	1893	5894	8543	9545	9882	9960	9989	9999
Cumulative Probability	0.0034	0.1893	0.5894	0.8543	0.9545	0.9882	0.996	0.9989	0.9999

TABLE 4 | *DOWN* STATE EQ(7)

Cumulative Density	<0.1	<0.2	<0.3	<0.4	<0.5	<0.6	<0.7
100 *simulation runs*							
Total count	0	5	19	66	90	100	100
Cumulative Probability	0	0.05	0.19	0.66	0.9	1	1
1000 *simulation runs*							
Total count	1	48	252	639	902	995	1000
Cumulative Probability	0.001	0.048	0.252	0.639	0.902	0.995	1
10,000 *simulation runs*							
Total count	13	364	2435	6161	9032	9899	10,000
Cumulative Probability	0.0013	0.0364	0.2435	0.6161	0.9032	0.9899	1

Sahinoglu Probability Model of Production Units (*Monte Carlo Simulation*) covering Eqs (9)–(24).

Plots shown in Figure 8 are the extrapolated JAVA versions of the EXCEL applications in Figures 5–7. Consequently a more detailed graphical JAVA version of the probability density plots with $n = 100{,}000$ simulation runs are displayed in Figures 9 and 10 to illustrate statistical centrality

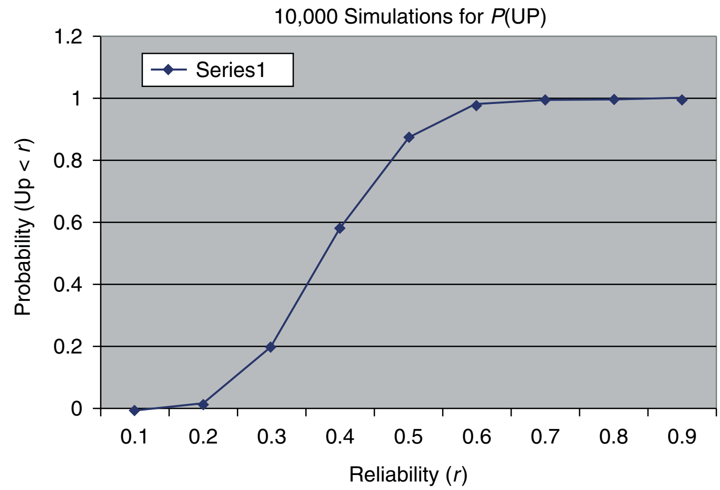

FIGURE 5 | *P*(*UP*) Cumulative reliability plot with 10,000 Monte Carlo simulation runs.

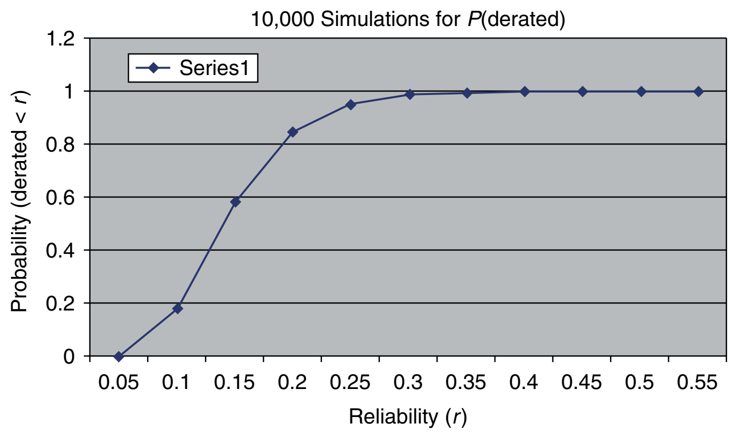

FIGURE 6 | *P*(*DER*) Cumulative reliability plot with 10,000 Monte Carlo simulation runs.

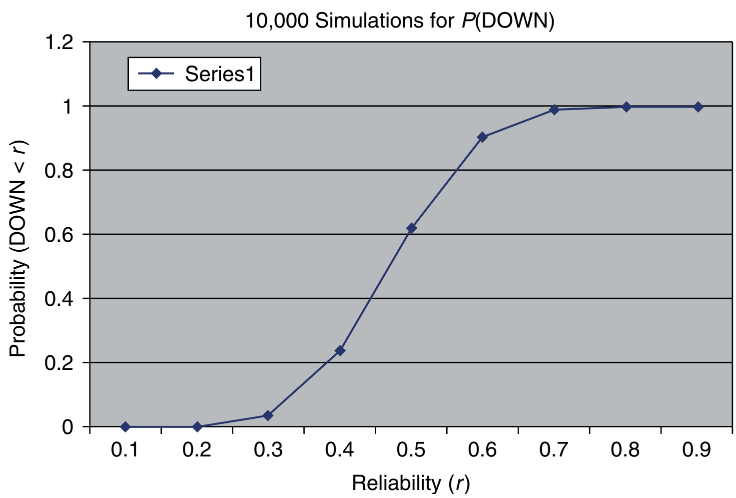

FIGURE 7 | *P*(*DOWN*) Cumulative reliability plot with 10,000 Monte Carlo simulation runs.

and location measures. The input data covering the first $n = 5$ events or episodes of each of six different sojourn times, as a hypothetical example in Table 1 are symbolically displayed in Figure 4, as derived from Markov state diagram shown in Figure 3. Consequently, results of Tables 2–4 and Figures 5–7 are plotted for each of the three states (UP–DOWN–DER) c.d.f. in Figure 8. Given the input tabulation in Table 1, the JAVA program will compute the popular statistical measures of three random variables as plotted in Figures 9 and 10. Probability density functions of the three states from Eqs (5)–(7) with a mean and a standard deviation, obtained by incremental piecewise calculations in

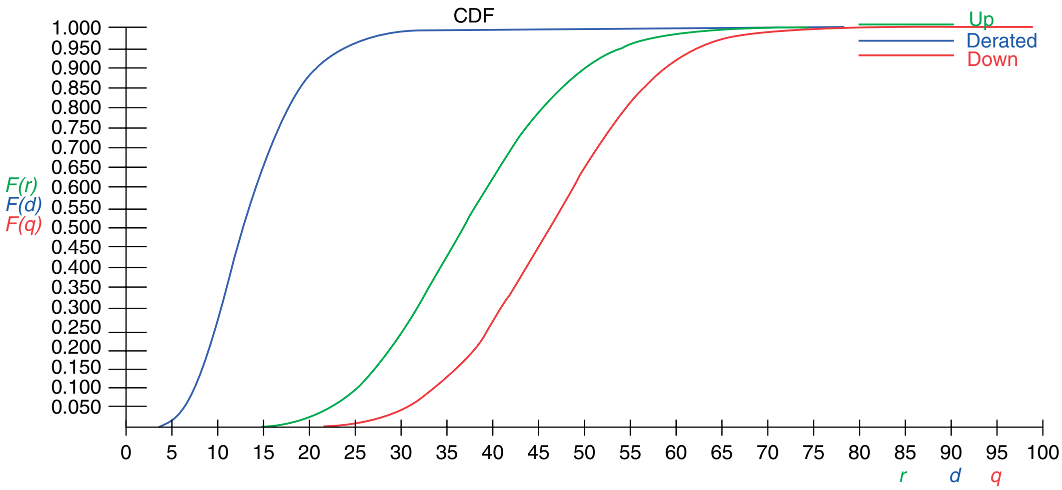

FIGURE 8 | The input data in Table 1, and simulation results in Tables 2–4 and Figures 5–7 display the cumulative reliability plots of the three states for *UP* (*r*), *DER* (*d*), and *DOWN* (*q*).

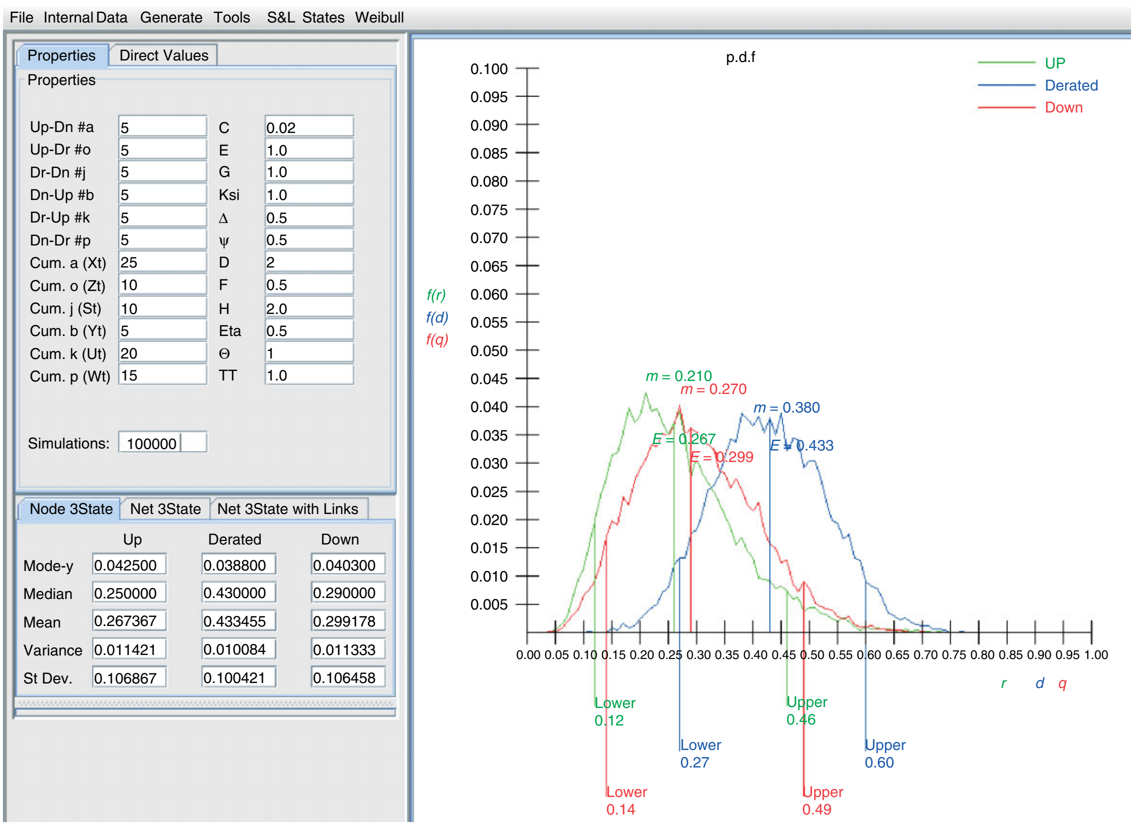

FIGURE 9 | Given the input table on the l.h.s. column, the p.d.f.s of the three states are plotted for *UP* (*r*), *DER* (*d*), and *DOWN* (*q*) for a 90% confidence level showing mode (*m*), mean (*E*) with upper & lower confidence as centrality measures for $n = 100{,}000$ simulation runs.

Figure 8 from the c.d.f.s of Figures 5–7 will follow:

$$f(UP) \sim Normal(0.267, 0.107),$$

$$f(DER) \sim Normal(0.433, 0.1)\ 46 \text{ and}$$

$$f(DOWN) \sim Normal(0.299, 0.106).$$

Note that the 90% confidence limits for the three Markov states computed in Figure 9 are as follow:

$\{UP_u = 0.12, \quad UP_L = 0.46\}$, $\{DER_u = 0.27, DER_L = 0.60\}$, $\{DOWN_u = 0.14, DOWN_L = 0.49\}$, respectively.

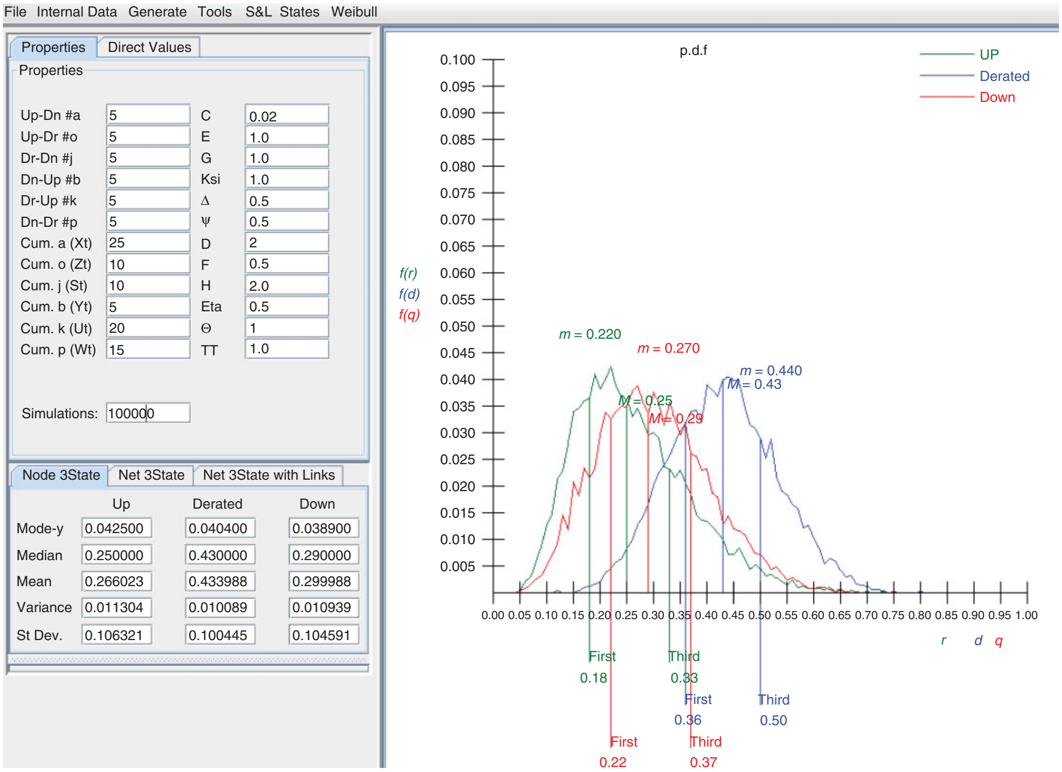

FIGURE 10 | Similar to Figure 12 but with Median (*M*), first and third quartiles as location measures for $n = 100{,}000$ simulation runs plotted for UP (*r*), *DER* (*d*), and *DOWN* (*q*).

Also note the first (Q_1) and third (Q_3) quartiles as location measures, computed in Figure 10 are as follow:

$\{UP_{Q1} = 0.18,\ UP_{Q3} = 0.33\}$, $\{DER_{Q1} = 0.36,\ DER_{Q3} = 0.50\}$, $\{DOWN_{Q1} = 0.22,\ DOWN_{Q3} = 0.37\}$, respectively.

Means $(E) \approx$ Medians $(M) \approx>$ Mode (m) for *UP*, *DER*, and *DOWN* are nearly identical. That is, E: $\{0.267, 0.433, 0.299\}$, M: $\{0.25, 0.43, 0.29\}$, and m: $\{0.21, 0.38, 0.27\}$ will result in a quasisymmetric plot. That is, Mean $(E) \approx$ Median (M) where a spike or two for the Mode (m) will not violate the symmetric appearance, as evident in Figures 9 and 10. Note, Mean $= E(q)$ and Median $= q_{0.5}$ if loss functions are assumed to be squared error and absolute error respectively, where mode is the maximum likelihood estimator. This follows from the fact that $E(q - \hat{q})^2$, if it exists, is a minimum when $\hat{q} = E(q)$, that is, the mean of the conditional (posterior) distribution of q. Then $E(q)$ is the Bayes solution:

$$E(q) = \int_0^1 q g_Q(q) \mathrm{d}q. \qquad (25)$$

Similarly according to Hogg and Craig[46] (p. 262), the median of the random variable Q is the Bayes estimator using an informative prior when the loss function is given as $L(q, \hat{q}) = |q - \hat{q}|$. If $E(|q - \hat{q}|)$ exists, then $\hat{q} = q_{0.5}$ minimizes the loss function, i.e., the median of the conditional posterior distribution of q. The median is resistant to changes. Then, $q_{0.5}$ or median of q, that is, q_M is the Bayes solution as the 50th percentile or 0.5 quantile, or second quartile for q, as follows:

$$0.5 = \int_0^{q_{0.5}} g_Q(q) \mathrm{d}q. \qquad (26)$$

How to Generate Random Numbers from Two-State Sahinoglu–Libby p.d.f. to Simulate Production Systems

Assume the random variables, $y \sim Gamma$ $(\alpha_1 = a + c,\ \beta_1 = \xi + x_T)$, and rv, $z \sim Gamma$ $(\alpha_2 = b + d,\ \beta_2 = \eta + y_T)$, where the random variable $q = y/(y + z)$ has the p.d.f. and c.d.f. respectively,

$$g_Q(q) = \frac{\Gamma(m' + n')}{\Gamma(m')\Gamma(n')} a'^{m'} b'^{n'} \frac{(1 - q)^{m'-1} q^{n'-1}}{[a' + q(b' - a')]^{m'+n'}}, \qquad (27)$$

$$G_Q(q) = 1 - G_{F_{2m',2n'}}\left[\frac{a'n'}{b'm'}(q^{-1} - 1)\right]$$
$$= P[F2m', 2n' > C_1 = (q^{-1} - 1)]. \quad (28)$$

Re-substituting for $n' = a + c$, $m' = b + d$, $b' = \xi + x_T$ and $a' = \eta + y_T$, we obtain for (27)

$$g_Q(\mathrm{q}) = \frac{\Gamma(a+b+c+d)}{\Gamma(a+c)\Gamma(b+d)}(\eta + \mathrm{y_T})^{\mathrm{b+d}}(\xi + \mathrm{x_T})^{\mathrm{a+c}} \times \frac{(1-q)^{b+d-1}q^{a+c-1}}{[\eta + y_T + q'(\xi + x_T - \eta - y_T)]^{a+b+c+d}}, \quad (29)$$

where Snedecor's *F*-Distribution used in Eq. (28) can be found in Ref 47. By the inverse transform approach, find the constant C_1 = inverse of $F2m', 2n'(1 - u_i)$ as in Eq. (28), by equating the c.d.f. value $G_Q(q)$ to a random uniform number, u_i for $i = 1, \ldots, N$ (large), as follows.

$$C_1 = \frac{a'n'}{b'm'}(q^{-1} - 1) \rightarrow q^*$$
$$= \frac{a'n'}{a'n' + C_1 b'm'}, 0 < q^* < 1, \quad (30)$$

where q^* is the *SL* ($\alpha = a + c$, $\beta = b + d$, $L = \beta_1/\beta_2$) random deviate for q (unavailability). Note, u_i are uniform (0,1) for $i = 1, \ldots, N$ (large). Figure 11 shows relationships between popular distributions for statistical simulations.

Example of *SL* Simulation for Modeling Network of 2 Two-State (UP-DN) Units

Given the following simplest series system of two identical components in Figure 12, whose default operational probability for each is $P(UP) = 0.9$ and hence $P(\text{System}) = 0.9^2 = 0.81$. We now force these units have their unavailability r.v. distributed with *SL* displayed as in Figure 2's l.h.s. column, where $g_Q(q)$ is formulated as follows: $SL(\alpha = a + c,\ \beta = b + d,\ L = (\xi + x_\mathrm{T})/(\eta + y_\mathrm{T})) = SL(\alpha = 10 + 0.02 = 10.02,\ \beta = 10 + 0.1 = 10.1;\ L = (1 + 1000)/(1 + 111.1) = (1001/112.1) = 9.7234)$. Use the *SL* random deviate simulator for q in Eq. (30), where q_i are to be independently *SL* distributed (Table 5). Historical failure and repair data are given in Figure 2. The flat deterministic outcome is $0.9^2 = 0.81$ whereas *SL*-distributional input-output relationship is unknown due to the closed-form derivation of the product of random variables being not available. Since Eqs (25) and (26) are not closed form solutions and tedious numerical integration is needed, Monte Carlo simulation can be the only solution for much larger networks if analytical tools are not available[45] (pp. 196–197, Figures 4 and 5) where analytical integration becomes an impossible task.

Example of Another *SL* Simulation for Modeling Network of 7 Two-State (UP-DN) Units

For the sake of a convenient example, a feasible and probable seven-node complex architectural style is taken[20] (p. 254) with failure and repair history including the prior parameters displayed on the l.h.s. with 10 each ups and downs lasting 1000 and 111.11 h, respectively, in Figure 13. The author assumes for the hypothetical control architecture an identical *SL*-distribution for its unavailability as displayed in Table 6 employing historical data for its components simulated 1000 times in 100-tuples of networks. This means 100,000 simulation runs overall. The analytical result being unknown for a complex system as the 7-node network depicted in Figure 13, the resulting simulation is 0.785 as in Table 6.

In the June 2012 issue of the IEEE Computer with 'Computing in Asia' as the cover feature, an article titled '*Computing for the Next-Generation Automobile*' displays three hybrid vehicle architectural styles: series, parallel and series-parallel, and then the (Toyota) Prius integrated THS II control architecture. It is mentioned that most vehicles today come with more than 50 embedded computer components, called electronic control units (ECUs)[7] (pp. 34–35).

A REVIEW OF MODELING AND SIMULATION IN CYBERSECURITY

Modeling and simulation (M&S) is a vital tool that can be leveraged for process improvement, and technology/capability development and evaluation. It is the process of designing a model of a system and conducting simulated experiments to preview and predict system behavior and evaluate optimal strategies for system operation. A short review of approaches will be covered in the world of cyber security on MC or DES. With the cyber security breaches rampant in the world, some of the most creative solutions to counteract these problems can be obtained by digital simulation faster, safer and cheaper than they can be resolved in the physical labs. In his related article, Rinaldi highlights M&S as a crosscutting initiative to increase the security of critical infrastructures.[48] Their *Strategy* states that modeling, simulation, and analysis must be employed to 'develop creative approaches and enable complex

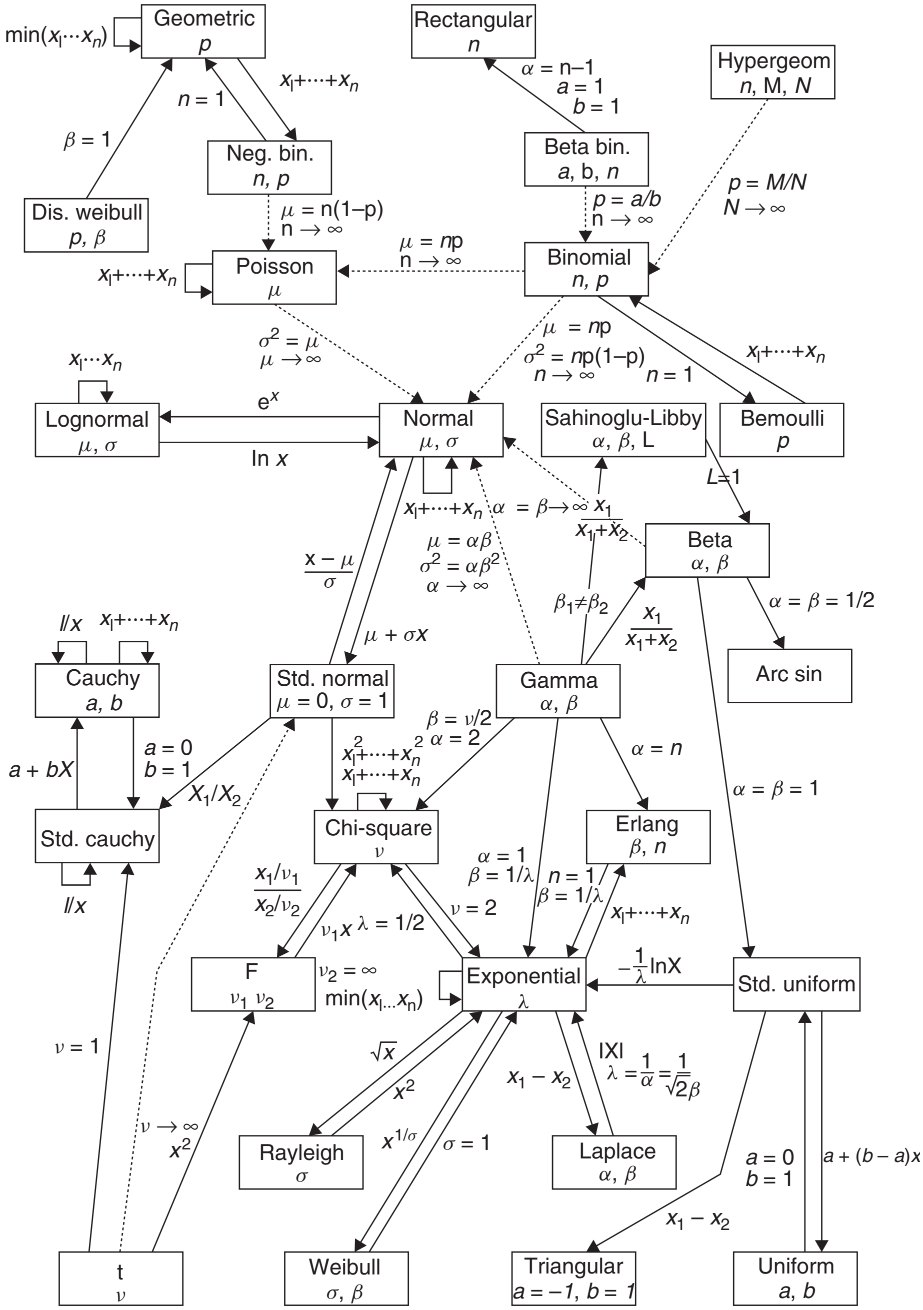

FIGURE 11 | Relationships for distributions in statistical simulation where $\alpha_1 = \alpha_2$ or $\alpha_1 \neq \alpha_2$, and $L = (\beta_1/\beta_2)$ for SL (α, β, L). (Dashed arrows indicate $\rightarrow \infty$ Reprinted with permission from Ref 20 Copyright 2007, Wiley & Sons, Inc)

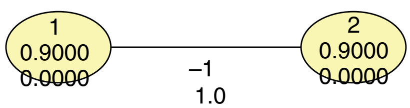

FIGURE 12 | Simple series system of two units.

decision support, risk management, and resource investment activities to combat terrorism at home'. Rinaldi concludes that the multidisciplinary science of interdependent infrastructures is immature, and requires M&S to mature it, and adds that they are developing, among others, at Sandia Labs the following techniques. Aggregate Supply and Demand (What-if Analyses), Dynamic Simulations, and ABM, which at a macro level similar to cellular automata, is out of scope for this review. Some examples of *M&S* and DES, in the cybersecurity field will follow.

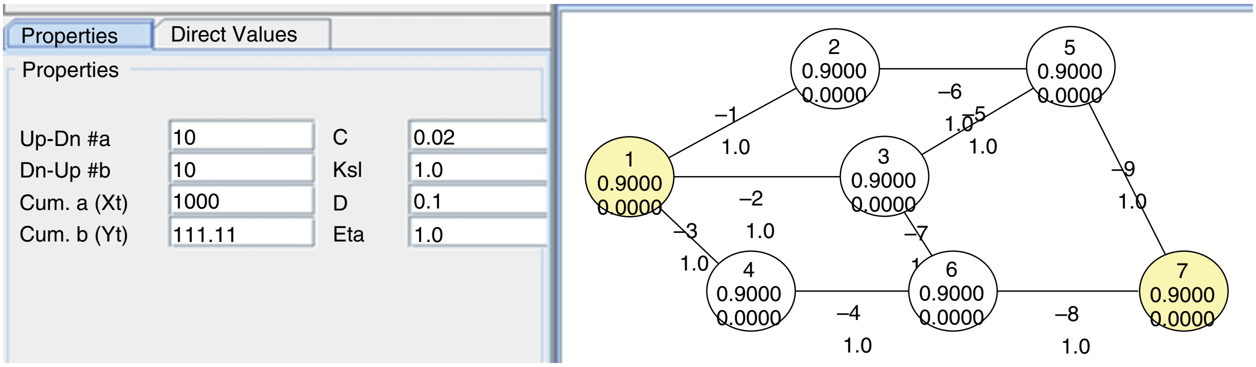

FIGURE 13 | Complex network of seven units with input data, where source: $s = 1$ and target: $t = 7$.

TABLE 5 | Simulation of a Simple Series Network using *SL*-Distributed Unit Unavailability in figure 12

79,617 successes out of 100,000 simulation runs. NETWORK RELIABILITY = 0.79617 NETWORK UNRELIABILITY = 0.20383 Each of the 100 networks simulated 1000 times totaling to 100,000 runs in 65.444 seconds

TABLE 6 | Simulation of a Complex Production Network using *SL*-Distributed Unit Unavailability in figure 13

78,476 successes out of 100,000 simulation runs. NETWORK RELIABILITY = 0.78476 NETWORK UNRELIABILITY = 0.21524 Each of the 100 networks simulated 1000 times in 148.36 seconds

Monte-Carlo Value-at-Risk Approach by Kim et al. in Cloud Computing

Based on today's volatile market conditions, the ability to generate accurate and timely risk measures has become critical to operating successfully, and necessary for survival. Value-at-Risk (VaR) is a market standard risk measure used by senior management and regulators to quantify the risk level of a firm's holdings. However, the time-critical nature and dynamic computational workloads of VaR applications make it essential for computing infrastructures to handle bursts in computing and storage resources needs. This requires on-demand scalability, dynamic provisioning, and the integration of distributed resources.

A VaR calculation will typically start after the end of the trading day, when market data and final positions have been verified. It must be complete, and updated risk numbers must be available, before the start of the next trading day. As the number and complexity of positions change, the computational requirements for the calculation can change significantly, however the completion deadline of the beginning of the next trading day remains fixed. Furthermore, as market conditions change, a firm may want to vary the number of Monte Carlo scenarios run (and thus the resolution of the calculation), which will add additional variability to the computation time. Specifically, the authors demonstrate how the Comet Cloud autonomic computing engine can support online multiresolution VaR analytics, a candidate for Cloud architecture by integrating of private and Internet cloud resources.[49]

Monte Carlo and Discrete Event Simulations in Sahinoglu's Security-Meter (SM) Risk Model

Four examples will be studied regarding Monte Carlo (MC) and Digital Event Simulation in the field of Cybersecurity.

Example for Security Meter Risk Modeling and Simulation

Assume two vulnerabilities and two threats in a $2 \times 2 \times 2$ set up as in Figure 14.[20,22]

Let X (total number of cyber-attacks detected) $= 360$/year and let $X_{11} = 98$, $X_{12} = 82$, $X_{21} = 82$, $X_{22} = 98$.

Let Y (total number of attacks undetected) $= 10$/year and let $Y_{11} = 2$, $Y_{12} = 3$, $Y_{21} = 3$, $Y_{22} = 2$.

When we keep Figure 14 in sight, we obtain the risk ratios and *ECL* (Expected Cost of Loss) as follow.

$$P_{11}(\text{threat 1 probability for vulnerability 1}) = (X_{11} + Y_{11})/(X_{11} + Y_{11} + X_{12} + Y_{12}) = \frac{100}{185} = 0.54, \quad (31)$$

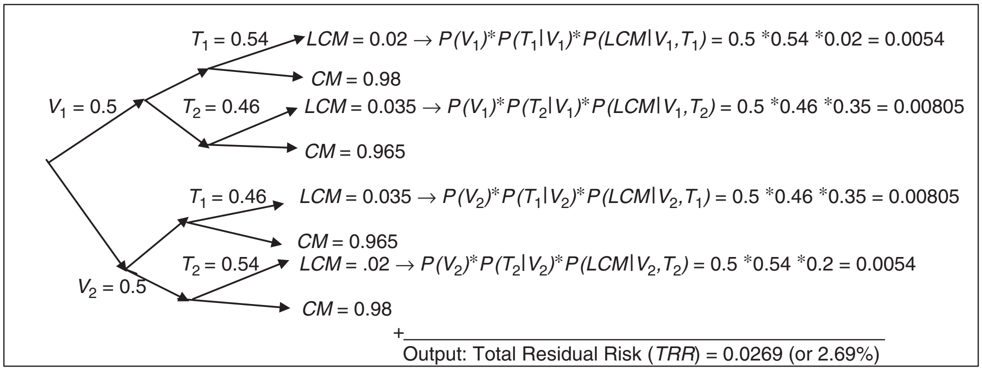

FIGURE 14 | Simplest 2 × 2 × 2 tree diagram for two threats and for two vulnerabilities in a cyber-risk scenario.

$$P_{12}(\text{threat 2 probability for vulnerability 1}) = (X_{12} + Y_{12})/(X_{11} + Y_{11} + X_{12} + Y_{12}) = \frac{85}{185} = 0.46, \quad (32)$$

$$P_{21}(\text{threat 1 probability for vulnerability 2}) = (X_{21} + Y_{21})/(X_{21} + Y_{21} + X_{22} + Y_{22}) = \frac{85}{185} = 0.46, \quad (33)$$

$$P_{22}(\text{threat 2 probability for vulnerability 2}) = (X_{22} + Y_{22})/(X_{21} + Y_{21} + X_{22} + Y_{22}) = \frac{100}{185} = 0.54, \quad (34)$$

$$P_1(\text{vulnerability 1}) = (X_{11} + Y_{11} + X_{12} + Y_{12})/(X_{11} + Y_{11} + X_{12} + Y_{12} + X_{21} + Y_{21} + X_{22} + Y_{22}) = \frac{185}{370} = 0.5, \quad (35)$$

$$P_2(\text{vulnerability 2}) = (X_{21} + Y_{21} + X_{22} + Y_{22})/(X_{11} + Y_{11} + X_{12} + Y_{12} + X_{21} + Y_{21} + X_{22} + Y_{22}) = \frac{185}{370} = 0.5. \quad (36)$$

The probabilities of LCM (Lack of Countermeasure) and CM (Countermeasure) where $CM + LCM = 1$ for the vulnerability-threat pairs demonstrated in Figure 14.

$$P(LCM_{11}) = (Y_{11})/(X_{11} + Y_{11}) = \frac{2}{100} = 0.02, \text{ hence, } P(CM_{11}) = 1 - 0.02 = 0.98, \quad (37)$$

$$P(LCM_{12}) = (Y_{12})/(X_{12} + Y_{12}) = \frac{3}{85} = 0.035, \text{ hence, } P(CM_{12}) = 1 - 0, \quad (38)$$

$$P(LCM_{21}) = (Y_{21})/(X_{21} + Y_{21}) = \frac{3}{85} = 0.035, \text{ hence, } P(CM_{21}) = 1 - 0.035 = 0.965, \quad (39)$$

$$P(LCM_{22}) = (Y_{22})/(X_{22} + Y_{22}) = \frac{2}{100} = 0.02, \text{ hence, } P(CM_{22}) = 1 - 0.02 = 0.98. \quad (40)$$

We place the estimated input values for the security meter in Figure 14 to calculate total residual risk.

Therefore, once you build the probabilistic model from the empirical data, as above, which should verify the final results, you can forecast or predict any 'taxonomic' activity whether it is the number of vulnerabilities or threats or crashes as in Table 7. For the study above, the total number of crashes is 10 out of 370 total events, which gives a ratio of $10/370 = 0.0270$ to verify the final results in Figure 14.

Using this probabilistically accurate model, we can predict what will happen in a different setting or year for a newly given explanatory set of data as in Table 7. If a clue suggests to us a future 1000 total episodes and 500 episodes of vulnerabilities of V_1, then by the avalanche effect, we can fill in all the other blanks, such as for $V_2 = 500$. Then (0.5405)(500) = 270.2 of T_1 and (0.4595)(500) = 229.7 of T_2. Out of 270.2T_1 episodes, (0.02)(270.2) = 5.4054 for LCM, were yielding to 5.4 crashes. Therefore, antivirus devices or firewalls have led to 264.8 preventions or saves. Again for T_2 of V_1: (0.035)(229.7) = 8.1 crashes and (0.965)(229.7) = 221.6 saves. The same holds for the V_2 due to symmetric data in this example depicted in Table 7. If the asset is \$2500 and the criticality constant is 0.4, then the ECL (expected cost of loss) is demonstrated in Figure 14 following above calculations. Also,

$$ECL = \text{Residual Risk} \times \text{Criticality} \times \text{Asset} = (0.0269)(0.4)(\$2500) = \$26.9. \quad (41)$$

TABLE 7 | The Deterministic Estimates of the *SM* Parameters in Figure 15 and Figure 16 Given the Total Number of Attacks

Total Attacks	VB	Attacks	%	Crashes	Saves	Threat	Events	%	Crashes	Saves	Risk	Post Pct	Post vb
370	v1	185	50.00	5	180	v1.t1	100.0	54.05	2.0	98.0	0.005405	20.00	0.0500000
						v1.t2	85.0	45.95	3.0	82.0	0.008108	30.00	0.0500000
	v2	185	50.00	5	180	v2.t1	100.0	54.05	2.0	98.0	0.005405	20.00	0.0500000
						v2.t2	85.0	45.95	3.0	82.0	0.008108	30.00	0.0500000
1000	v1	500	50.00	14	486	v1.t1	270.2...	54.05	5.4054...	264.8...	0.005405	20.00	0.0500000
						v1.t2	229.7...	45.95	8.1081...	221.6...	0.008108	30.00	0.0500000
	v2	500	50.00	14	486	v2.t1	270.2...	54.05	5.4054...	264.8...	0.005405	20.00	0.0500000
						v2.t2	229.7...	45.95	8.1081...	221.6...	0.008108	30.00	0.0500000

Discrete Event (Dynamic Time-Dependent) Simulation using Negative Exponential p.d.f.

The analyst is expected to simulate a cyber-component's (such as a server) tree-diagram 10 consecutive times from the beginning of the year (e.g., 1/1/2013) until the end of 1000 years (i.e., 12/31/3012) in an 8,760,000 h period, with a life cycle of crashes or saves for a total of $10 \times 1000 = 10{,}000$ simulation runs. The input data is tabulated in Table 7 to conduct the generation of random deviates. At the end of this planned time period, the analyst will fill in the elements of the tree diagram for a $2 \times 2 \times 2$ security meter's tree diagram model as in Figure 14. Recall that the rates are the reciprocals of the means for the assumption of a *negative exponential probability density function* to represent the distribution of time to crash. For example, if $\lambda = 98$ per 8760 h, the mean time to crash is $8760/98 = 89.38$ h. Use the input as in Table 7.[20,22] We observe a result of $TRR = 0.0269 \approx 0.027$ in Figure 15.

Monte Carlo (Static Time-Independent) Simulation using Poisson p.d.f.

Using the identical information in Section *Example for Security Meter Risk Modeling and Simulation*, the analyst is expected to use the principles of Monte Carlo simulation to simulate the $2 \times 2 \times 2$ security meter as in Table 7 and Figure 14 for 10 repeated trials. One employs the Poisson distribution for generating failure and repair rates for each leg in the tree diagram of the $2 \times 2 \times 2$ model shown in Figure 14. The rates are given as the count of saves (repairs) or crashes (failures) annually. The necessary rates of occurrence for the Poisson distribution's random value generation were given in the empirical data in Table 7 above. For each security meter realization, get a risk value and average it over $n = 10{,}000$ in 1000 increments. When you average over $n = 1000$ runs, you should get the same value as in Figure 15. Using the same data, as projected, we get the same results in Figure 16 as in Figure 15. That is, $TRR = 0.0269 \approx 0.027$.[20,24] Therefore, DES and MC results were identical to four decimals as expected.

Monte Carlo (Static Time-Independent) Simulation using a Continuous Uniform p.d.f.

In another cyber server setting, with three vulnerabilities (V_1, V_2, V_3) with four threat levels for V_1, three threat levels each for V_2 and for V_3; the following upper and lower risk values for $U(a,b)$ are assumed to be available for each vulnerability, threat and corresponding *LCM* variables, as follow in Table 8. Selected upper and lower uniformly distributed example values are very close to reduce variation. Theoretical derivations for *TRR*'s mean and variance are by MAPLE software using are as follow.[24]

$M := 0.260432113$, $V := 0.0000144453852$, as copied from MAPLE outcomes, are tabulated in Table 8:

$$\begin{aligned} M := {} & 0.05040004790 + 0.03247999996 \\ & + 0.003360000913 + 0.002800000577 \\ & + 0.03717999068 + 0.003380000475 \\ & + 0.007903999736 + 0.03494401504 \\ & + 0.06903005201 + 0.01895400665. \end{aligned}$$

Tables 9 and Figure 17 provide the comparative analytical MAPLE tabulations and Monte Carlo Simulations below. The means are almost identical to 10^{-6} identical and standard deviations are only 9×10^{-5} apart. This alone shows the predictive power of modeling and simulation for cyber security studies in Section *A Review of Modeling and Simulation in Cybersecurity* in addition to those of production or manufacturing engineering in Section *A Cross Section of Modeling and Simulation Issues in Manufacturing*. Therefore, $ECL = \text{Final Risk} \times \$8\text{K} = 0.26 \times 0.4 \times$

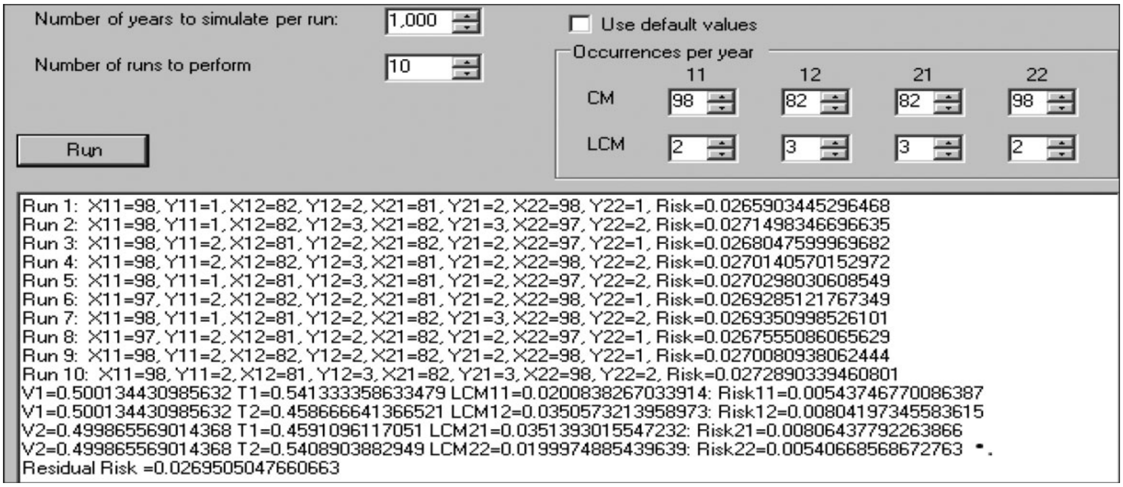

FIGURE 15 | DES results of the 2 × 2 × 2 security meter sampling design.

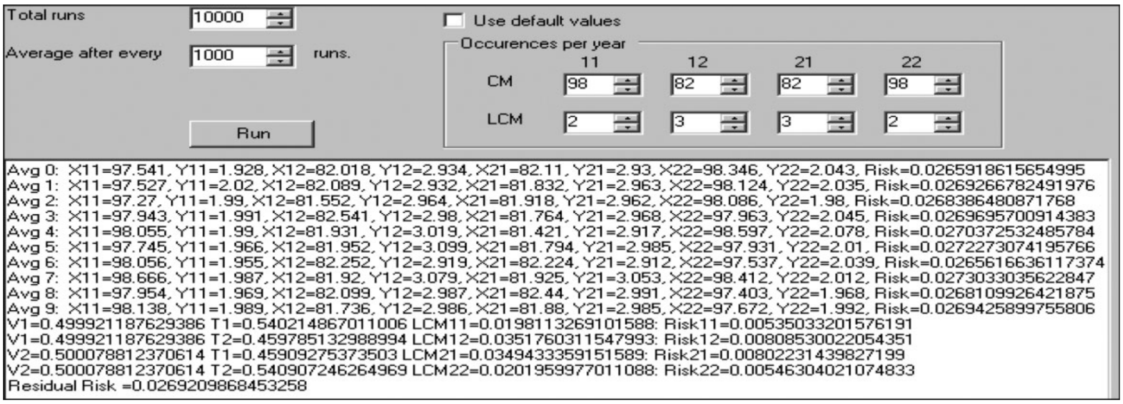

FIGURE 16 | The Monte Carlo (MC) simulation results of the 2 × 2 × 2 security meter sampling design.

TABLE 8 | Simulation Input Data for the SM's Uniformly Distributed $U(a, b)$, $TRR = \sum_1^{10} \text{RRi} = 0.26$

Vulnerability		Threat		Lack of Countermeasure		Residual Risk (RR_i)
Lower	Upper	Lower	Upper	Lower	Upper	Expected
0.34	0.36	0.47	0.49	0.29	0.31	0.0504
		0.15	0.17	0.57	0.59	0.0324
		0.31	0.33	0.02	0.04	0.0033
		0.03	0.05	0.19	0.21	0.0028
0.25	0.27	0.21	0.23	0.64	0.66	0.0371
		0.01	0.03	0.64	0.66	0.0033
		0.75	0.77	0.03	0.05	0.0079
0.38	0.40	0.31	0.33	0.27	0.29	0.0349
		0.58	0.60	0.29	0.31	0.0690
		0.08	0.10	0.53	0.55	0.0189

\$8K = 0.104173 × \$8K = \$833.38 is the expected cost of loss to redeem if no risk is desired. How to mitigate the accrued risk from unwanted to a tolerable risk percentage is detailed in Refs 24 and 50.

DISCUSSION AND CONCLUSION

The power of simulation is evident from countless number of contemporary research works in addition to industrial and military undertakings to save time

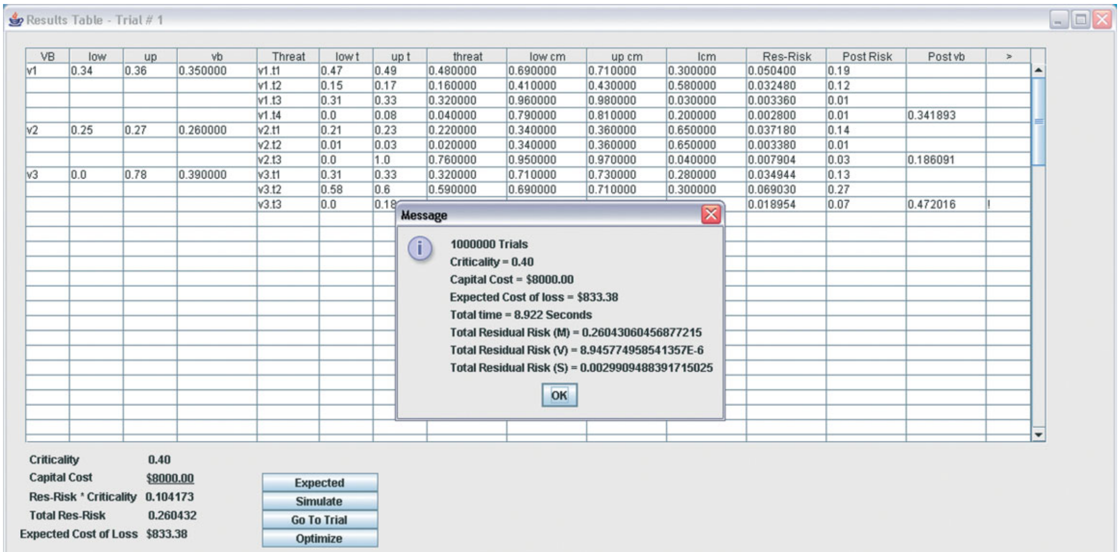

FIGURE 17 | Monte Carlo simulations for the cyber server ($8000 asset) example with inputs in Table 8.

TABLE 9 | Comparison of Monte Carlo and Analytical Results for the Cyber Server from Input of Table 8

Monte Carlo Simulation for $U(a,b)$ in 1 M runs ($n = 1{,}000{,}000$)	MAPLE (analytical) for $U(a,b)$
$TRR(M) = 0.260430$	$TRR(M) = 0.260432$
$TRR(V) = 0.0000089$	$TRR(V) = 0.00001444$
$TRR(S) = 0.0029$	$TRR(S) = 0.0038$

and budget. Besides nondestructively 'learning the truth' before 'unexpected things happen' in the real-world sense at an incomparably cost effective setting; the science and art of *M&S* cracks the code for numerous challenging problems where analytical derivations or formulae prove inutile by reaching a dead-end. The objective of applying simulation is to strengthen the advantages of the IT corporate circles and reduce the disadvantages, mainly because of the economic pressure and time constraints in the business world. A gamut of modeling and simulation practices in the Armed Forces flank can be advantageously utilized to plan saving time and resources so as to avoid wasting a tight budget for 'the most bang for the buck' before new projects are hastily commissioned, only to see that they are not what to get the job done in a disappointing finale. Uses of simulation in medical oncology or else, as well as its impact in the area of computational finance are only some of its virtually endless applications.[51,52] Following a brief introduction and running a best-kept-secret historical perspective to the origins of simulation, the author reviews the literature as to why the art and science of modeling and simulation are crucial to today's engineering world. The review further focuses on the currently popular manufacturing and cyber defense issues, to cite a few examples if not all, to set the stage for the rest of the plentiful engineering avenues.

On the manufacturing or production front, the author in response to then-in-2007-unsolved homework question 5.5 on p. 256 from his Wiley textbook,[20] reviews the set of techniques to generate the multistate probability distribution model of an important pillar of trustworthiness, that is, availability. Namely, when the availability (or reliability) of a unit is at stake, and while the unit possesses three operational states with a derated state added beyond the usual two-state binary or dichotomous assumption, conventional applications do not suffice. Therefore, it is worth to review the fact that the primary difference between other related works[30–32] and author's empirical Bayesian treatment of the three states of a repairable hardware unit is to estimate the p.d.f.s of these three states by using Monte Carlo simulations.[38,39] The closest article to this one uses only four transition rates in a three-state Markov model whereas the reviewed Monte Carlo model uses all six transitions.[30] This reviewed statistical simulation approach is powerful and flexible, whereas Ref 30 deals with differential equations limited in scope. Other close references deal with different topics; however none use any simulation techniques.[31,32]

It is currently infeasible to find closed form solutions for the random variables of *UP*, *DER*, and *DOWN* expressed by Eqs (5)–(8) because of a multiplicity of sums and products of *gamma* random variables expressed in the denominator term of *Three-State Sahinoglu Probability Model of Production Units* (*Monte Carlo Simulation*). In the final analysis shown in *Statistical Simulation of Three-State Units to Estimate the Density of UP*, *DOWN and DER*, the resulting distributions for the three parameters, *UP*, *DER* and *DOWN* are approximated by normal distributions. The outcome distributions in *A Cross Section of Modeling and Simulation Issues in Manufacturing* are quasisymmetrical with *E* (Mean) and *M* (median) almost equal, although slightly right-skewed because Mean $\approx$ Median $>$ Mode. The reviewed Sahinoglu–Libby, a.k.a., *SL* (α, β, L) is the continuous probability density function of the unavailability (or availability when duly reparametrized) of a two-state unit. For those units whose life time can be decomposed into operating (*UP*), derated (*DER*), and nonoperating (*DOWN*) states in a three-state setting, sojourn times are assumed to be distributed according to the generalized *gamma* p.d.f. where both shape and scale parameters are non-identical. The resultant density plots in Figures 12 and 13, following extensive statistical simulations in *A Cross Section of Modeling and Simulation Issues in Manufacturing*, are approximately symmetric normal despite a spike for the mode. These plots definitely qualify to pass goodness-of-fit tested for Normal p.d.f. Because of infeasibility of closed form analytical solutions for the explained three-state version; the Monte Carlo simulation technique is rightfully selected as a mathematically tractable model to calculate the *UP*, *DER*, and *DOWN* probabilities for a three-state repairable hardware unit.

These summary measures are all shown in the plots of the JAVA applications throughout *A Cross Section of Modeling and Simulation Issues in Manufacturing*. Network applications for medium and large networks are studied using Monte Carlo simulations in references.[20–24] After the analyses, the approximate closed form p.d.f.s can be estimated as shown in *Statistical Simulation of Three-State Units to Estimate the Density of UP*, *DOWN*, *and DER* because of the favorable results by normal probability plots. Researchers can utilize the results for their related research when deriving the p.d.f.s of their Markov states in other disciplines such as business, for example, banking.[53] Currently, only deterministic probabilities can be calculated through Markov algebra, but not their probability densities. For example, a credit card is either closed (if less than a critical credit score), open (more than) or only conditionally usable for urgent cases (between lower and upper). The Bank actuarians may want to estimate the p.d.f.s of these three states to conduct statistical inference using customer-based empirical data by employing empirical Bayesian analysis. Multi-state systems such as in the case of four multiple derated states representing electric power turbines, as cited in Refs 20 (p. 280, Figure 6.26) and 45 (p. 201, Figure 10) can be derived. These estimators for unit availability can further be propagated to simulate the source-target availability for troublesome complex networks.

Regarding the cyber security science and engineering issues however, implementation of modeling and simulation compared to manufacturing industry is fairly new progressing at an experimental stage. This fact is not only because of involvement of human life and death situations in adversity, as compared to accidental casualties in the production world, but also because of lack of theoretical and experiential data base dating back to only 1990s since the launch of public internet. The author, by following examples in this area proceeds with currently popular VaR technique by Kim et al.[49] and Security Meter and CLOUD simulation tools (CLOURA) by Sahinoglu et al.[26] Monte Carlo VaR is costly to execute; it does not incorporate cost comparisons when taking measures. Consider a medium size firm holding positions in 20,000 different financial instruments. Running a 100,000 simulation Monte Carlo VaR calculation requires generating 2 billion simulated instrument prices. With a conservative estimate of 10 milliseconds per pricing, this calculation requires more than 5500 h of processor time over an 8 h window. The capital cost of hardware plus the operational cost for data center space, power, cooling and maintenance makes this cost prohibitive to all but the largest firms. However, scalable CLOURA is a very fast algorithm, that is, it can simulate a CLOUD system with 430 servers for 1000 years in less than 4 min.[26] *SM* simulations as in Tables 7–9 and Figures 15–17 and are relatively fast and accurate, comparable to their analytical counterparts.[20–22]

Overall M&S techniques abound, particularly face-saving in the case of theoretical impasses, and sometimes the only viable solutions in engineering and scientific applications. The multiples of positive results render M&S methods among the most useful and practical, as well as affordable algorithms of our time. If one day, humankind can make it to the surface of the red planet Mars, it will be possible because humans will have nondestructively travelled to Mars some tri-zillion times by riding on the cyber space through digital simulation rather than on the outer space. The author contends that positive solutions will realize for cancer and currently incurable diseases by crunching computationally intensive and nonlethal M&S techniques. The application of M&S to engineering, cyberspace and health informatics, however, is not an easy task with much progress remains to be done. This overview also aims to provoke thoughts and stimulate ideas for such goals by exploring interdisciplinary avenues through M&S using supercomputing.

Finally, one exam question in a Cybersecurity M.S. program's midterm exam at AUM[54] asked, 'What would separate you in your future job if you took an M&S course, and others did not have any clue?' The following four responses in Appendix B were gathered from candidates invariably all with a military background, either on active duty or retired USAF. The responses, as quoted, did demonstrate an awakening of mind on the timely significance of Modeling and Simulation in cyberspace and reliability and security engineering.

ACKNOWLEDGMENTS

I wish to acknowledge the late Distinguished Professor Norman Lloyd Johnson, the recipient of the Wilks Award and Shewhart Medal, and a Fellow of the Institute of Actuaries. NLJ was one of the finest old-school gentleman scientists I knew although I never had a chance to meet with him in person other than call and e-mail. His invaluable advice to the originators of the *SL* math-statistical derivation, D.L. Libby and me, was '...emphasizing applications which I regard as the more important feature...' This is what this review article targets to achieve in his memory. When I learned of his passing at Sydney's ISI/2005 from a renowned statistician, his compatriot, Sir David Cox, I was *very* saddened. The author regrets not having corresponded with NLJ after leaving CWRU (1999) where NLJ was earlier a visiting scholar (1960–61) at the Case Institute of Technology. Prof. Johnson's uniquely handwritten letters and one-typed letter to Drs. Libby and Sahinoglu are of historical value. Dr. Johnson added in his last letter, dated July 9, 2000, and addressed to the author while at TSUM that he would suggest to his coauthors, Kotz and Balakrishnan, an addition in a new book years after his first,[55] in referring to both Ph.D. dissertations and the naming of the distribution, as such, in his third letter. In his earlier hand-written letter dated April 6, 1999 while I was at CWRU, Professor Johnson stated, 'Dr. Libby and yourself were, it appears, first on parallel courses, both in regards to PhD dissertation (in 1981) and in later publications of FOR (yours) and G3B (his) dissertations and their derivatives.' Dr. Johnson's support for younger statisticians was exemplary. May Dr. NLJ (1917–2004) rest in peace.

Additionally, I wish to acknowledge Roy Billinton (Professor Emeritus, earlier from University of Saskatchewan, Saskatoon, Canada) who instructed me in 1974–1975 my first course on Power Reliability Engineering at the University of Manchester's (UK) Institute of Science and Technology (UMIST). Roy, an immigrant from Leeds to Canada in 1950s is considered to be the founder of the modern Power Systems Reliability. His book that I quoted in this review dating back to 1970s was the first such source for me in the field. His rigor and love of teaching led me to a pursuit of reliability engineering. I too acknowledge Dr. R. Allan, my thesis supervisor from UMIST. Later, during my PhD studies at Texas A&M (1977–1981), other scholars who were instrumental to my formative education related to this article's content are; all Prof. Emeriti, A.D. Patton, A.K. Ayoub (deceased), C. Singh from EE Department and L. Ringer, M. Longnecker, H.O. Hartley (deceased), O. Jenkins and C. Gates, from whom I [illegible] learned random numbers. Yuan Y. Ling, S. Capar, D. Tyson, S. Escue, and from AUM's CSIS, Messieurs Kramer, Pelkey, Kelsoe, Stockton, Mims, Vasudev, Morton, Zhang and Samelo, who contributed to the various editing, programming and interviewing stages. I am indebted to the anonymous reviewers, such as those I listed in the ScholarOne when asked for WIREs, that is, Drs. Iyengar (FIU), Valenzuela (AUBURN), Simmons (UNCW) and Phoha (LATECH) who meticulously improved this article's readability. Finally over four years of my scholarly exchange of academic expertise since mid-2008, those WIREs warriors—who did so much to succeed—I must acknowledge are, Drs. E. Wegman, Y. Said, and D. Scott; while admiring what they overachieved to make COMPSTATS a repeat award winning journal with C. Strickland, Assoc. Editor, kind and omnipresent to help resolve issues invariably. WIREs series ought to be also congratulated for their service to the computational science.

REFERENCES

1. Available at: http://www.nucleonica.net/wiki/images/d/df/EMC_2008_short.pdf. (Accessed January 5, 2013).
2. Available at: http://web.student.tuwien.ac.at/~e9527412/history.html. (Accessed January 5, 2013).
3. Available at: http://www.britannica.com/EBchecked/topic/1252440/von-Neumann-machine. (Accessed January 5, 2013).
4. Available at: http://mathworld.wolfram.com/CellularAutomaton.html. (Accessed January 5, 2013).
5. Available at: http://www.math.com/students/wonders/life/life.html. (Accessed January 5, 2013).
6. Ledin J. *Simulation Engineering- Build Better Embedded Systems Faster*. Taylor and Francis; CRC Press; BOCA Raton, London & Newyork 2001. Available at: http://books.google.com/books/about/Simulation_Engineering.html?id=GMRjR2shhXAC. (Accessed January 5, 2013)
7. Aoyama M, Computing for the next-generation automobile. *IEEE Comput* 2012, 45:32–37.
8. Zeigler BP, Praehofer H, Kim TG, *Theory of Modeling and Simulation*, 2nd ed. Academic Press San Diego, CA 92101-4495,USA; 2000.
9. Levene H, Wolfowitz J. The covariance matrix of runs up and down. *Ann Mat Stat (AMS)* 1944, 15:58–69
10. Knuth DE, *The Art of Computer Programming, Vol. 2: Seminumerical Algorithms*. 3rd ed. Addison-Wesley: Reading, MA; 1998.
11. Stewart WJ. *Probability, Markov Chains, Queues and Simulation*. Princeton University Press Princeton, NJ, 08540, USA; 2009.
12. Available at: http://www.youtube.com/watch?v=_RBH0PeLhOk. (Accessed January 5, 2013).
13. Available at: http://www.nap.edu/openbook.php?record_id = 10425&page = 77. (Accessed January 5, 2013).
14. Available at: http://www.guardian.co.uk/public-leaders-network/2011/may/20/local-councils-software-hard-cuts. (Accessed January 5, 2013).
15. Available at: http://www.albrechts.com/mike/DES/Annotated%20Bibliography%20.pdf. (Accessed January 5, 2013).
16. Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=373979 (Accessed January 5, 2013).
17. Kuhl E, Kistner J, Costantini K, Sudit M. Cyber attack modeling and simulation for network security analysis. *ACM Digital Library*, *Proceedings of the 39th Conference on Winter Simulation Conference* (*WSC'07*), 2007.
18. Available at: http://www.youtube.com/watch?v = du6g__lgS3Q (Accessed January 5, 2013).
19. Available at: http://www.washingtonpost.com/wp-dyn/content/article/2010/02/16/AR2010021605762.html (Accessed January 5, 2013).
20. Sahinoglu M. *Trustworthy Computing: Analytical and Quantitative Engineering Evaluation* (CD ROM included). Hoboken, NJ: John Wiley & Sons Inc.; 2007.
21. Sahinoglu M. Security meter—a practical decision tree model to quantify risk. *IEEE Securit Privacy* 2005, 3:18–24.
22. Sahinoglu M. An input-output measurable design for the security meter model to quantify and manage software security risk. *IEEE Trans Instrum Measure* 2008, 57:1251–1260.
23. Sahinoglu M. Can we quantitatively assess and manage risk of software privacy breaches?. *Int J Comput, Inform Technol Eng* 2009, 3:65–70.
24. Sahinoglu M. Yuan Y-L. Banks D. Validation of a security and privacy risk metric using triple uniform product rule. *Int J Comput Inform Technol Eng* 2010, 4:125–135.
25. This file appeared originally in the Wikipedia article entitled Computer Simulation. It is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported License. Available at: http://en.wikipedia.org/wiki/File:Molecular_simulation_process.svg. (Accessed January 5, 2013).
26. Sahinoglu M, Cueva-Parra L. CLOUD computing. *Wiley Interdisciplinary Reviews Computational Statistics* 2011, 3:47–68.
27. Jain S, Foley WJ. Basis for development of a generic FMS simulator. *Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*. Amsterdam: Elsevier Science Publishers B.V.; 1986, 393–403.
28. Schriber TJ, Stecke KE. Machine utilizations and production rates achieved by using balanced aggregate FMS production ratios in a simulated setting. *Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*. Amsterdam: Elsevier Science Publishers B.V.; 1986, 405–416.
29. Dee ZJ, Co HC, Wyek RA. SIM-Q: a simplified approach to simulation modelling of automated manufacturing systems. *Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*. Amsterdam: Elsevier Science Publishers B.V.; 1986, 417–430.
30. Rao MS, Naikan VN, Pradesh A. A hybrid markov system dynamics approach for availability analysis of degraded systems. *Proceedings of the 2011 International Conference on Industrial Engineering and Operations Management*, Kula Lumpur, Malaysia, January 22–24, 2011.

31. Lins ID, Droguett EL. Multiobjective optimization of availability and cost in repairable systems design via genetic algorithms and discrete event simulation *Pesqui. Oper*. Jan/Apr 2009, **29**.
32. Shah A, Dhillon BS. Reliability and availability analysis of three-state device redundant systems with human errors and common-cause failures. *Int J Perform Eng* 2007, 3:411–441.
33. Sahinoglu M, Longnecker MT, Ringer LJ, Singh C, Ayoub AK. Probability distribution function for generation reliability indices-analytical approach. *IEEE Trans Power Apparat Syst (PAS)* 1983, 104:1486–1493.
34. Sahinoglu M, Libby D, Das SR. Measuring availability indices with small samples for component and network reliability using the Sahinoglu-Libby probability model. *IEEE Trans Instrum Measure* 2005, 54:1283–1295.
35. Sahinoglu M. Statistical inference on reliability performance index for electric power generation systems. Ph.D. Dissertation, The Institute of Statistics jointly with Electrical and Computer Eng., Texas A&M University, College Station, December 11, 1981.
36. Libby DL. Multivariate fixed state utility assessment. Ph.D. Dissertation, University of Iowa, Iowa City, 1981.
37. Libby DL, Novick MR. Multivariate generalized β distributions with applications to utility assessment. *J Ed Stat* 1982, 7:271–294.
38. Sahinoglu M, Yuan Y-L, Capar S. Statistical inference on the two- and three- state availability of the repairable units with the Sahinoglu-Libby Model, IPS018 (Invited Paper Session): statistical risk assessment in trustworthy computing. Proceedings (CD ROM) of International Statistical Institute, 58th Congress, Dublin, 2011.
39. Sahinoglu M, Yuan Y-L. Multivariate statistical modeling on the 3-state (up, derated, down) availability of repairable hardware unit and their systems with the Sahinoglu-Libby probability distribution using monte carlo simulation. *Proceedings of GCMS'*10, Ottawa, Canada, 12–14 July 2010, 253–260.
40. Lisniansky A, Levitin G. *Multi-State System Reliability: Assessment, Optimization and Applications*. World Scientific Publishing Co Ptc.Ltd, Singapore 596224; 2003.
41. Billinton R, Allan R. *Reliability Evaluation of Power Systems*. New York: Plenum Press; 1996.
42. Murchland J. Fundamental concepts and relations for reliability analysis of multistate systems, reliability and fault tree analysis. *Theoret Appl Aspects System Reliab SIAM* 1975:581–618.
43. Barlow R, Wu A. Coherent systems with multistate elements. *Math Oper Res* 1978, 3(4):275–281.
44. Billinton R. *Power System Reliability Evaluation*. One Park Avenue, New York, NY: Gordon and Breach Science Publishers Inc. 1970.
45. Sahinoglu M, Rice B. Network reliability evaluation. *Wiley Interdiscip Rev: Comput Stat* 2010, 2:189–211.
46. Hogg RV, Craig AT. *Introduction to Mathematical Statistics*. 3rd ed. New York: MacMillan; 1970.
47. Snedecor G, Cochran W. *Statistical Methods*. Ames, IA: Iowa State University Press; 1989.
48. Rinaldi SM. Modeling and simulating critical infrastructures and their interdependencies. *IEEE Proceedings of the 37th Hawaii International Conference on System Sciences*, Hawaii, 2004, 1–8
49. Kim H, Chaudhuri S, Parashar M, Marty C. Online risk analytics on the cloud. IEEE Computer Society Washington, DC, CCGRID'09. *Proceedings of the* 2009 *9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, 2009, 484–489.
50. Sahinoglu M, Cueva-Parra L, Ang D. Game-theoretic computing in risk analysis. *WIREs Comp Stat* 2012, 4:227–248. doi:10.1002/wics.1205.
51. Thompson J. Forward simulation models. *Wiley Interdiscip Rev Comput Stat* 2010, 2:61–68.
52. Kroese DP, Rubinstein RV. Monte Carlo methods. *WIREs Comp Stat* 2012, 4:48. doi:10.1002/wics.194
53. Underwood RG. A non-trivial markov chain with explicit invariant distribution. Available at: http://sciences-srv.aum.edu/~runderwo/papers/Markov_paper_underwood.pdf. (Accessed January 28, 2013).
54. Available at: www.aum.edu/csis. (Accessed January 5, 2013).
55. Johnson NL. Kotz S. *Distributions in Statistics: Continuous Univariate Distributions*. Wiley-Interscience Newyork, USA; 1970.

APPENDIX A

See Figures A1–A9.

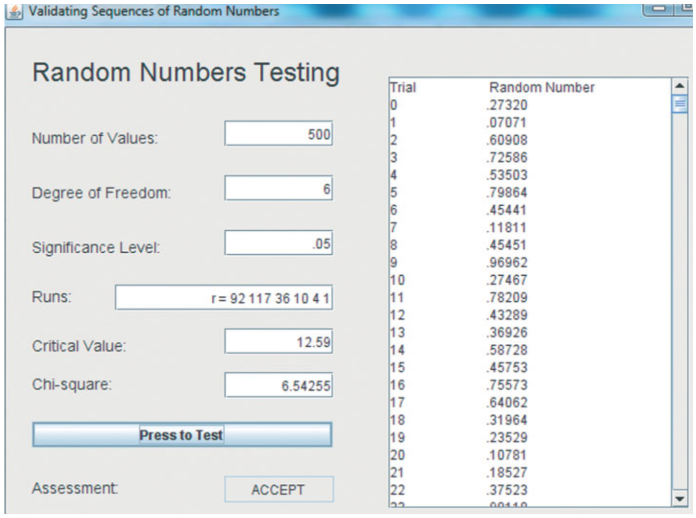

FIGURE A1 | Uniform numbers testing; Ho: random versus Ha: not random for 500 runs. Ho is not rejected.

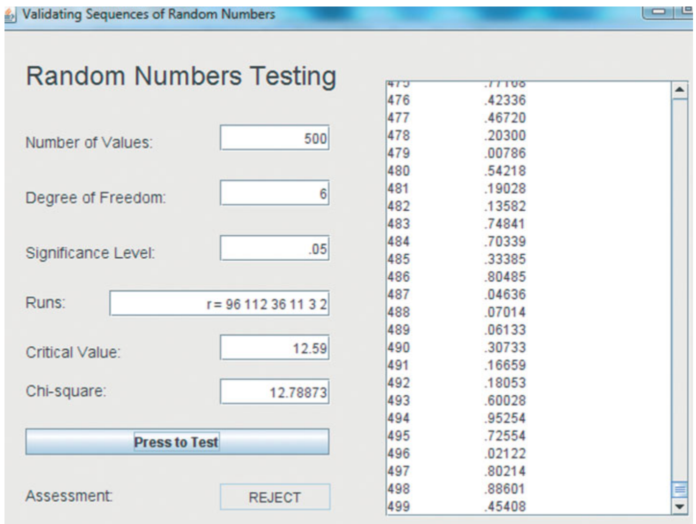

FIGURE A2 | Uniform numbers testing; Ho: random versus Ha: not random for 500 runs. Ho is rejected. On the average, one out of 40 cycles of 500 runs = 20,000 simulations will end up rejecting Ho: random.

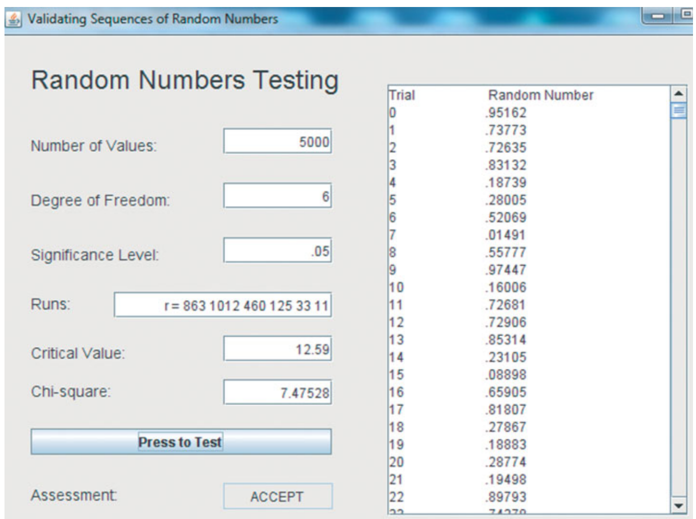

FIGURE A3 | Uniform numbers testing; Ho: random versus Ha: not random for 5000 runs. Ho is NOT rejected.

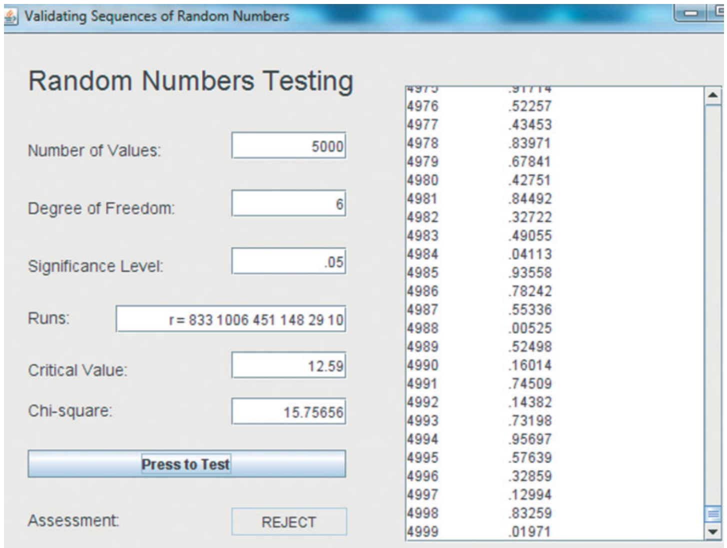

FIGURE A4 | Uniform numbers testing; Ho: random versus Ha: not random for 5000 runs. Ho is rejected. On the average, one out of 10 cycles of 5000 = 50,000 simulations will end up rejecting Ho: random.

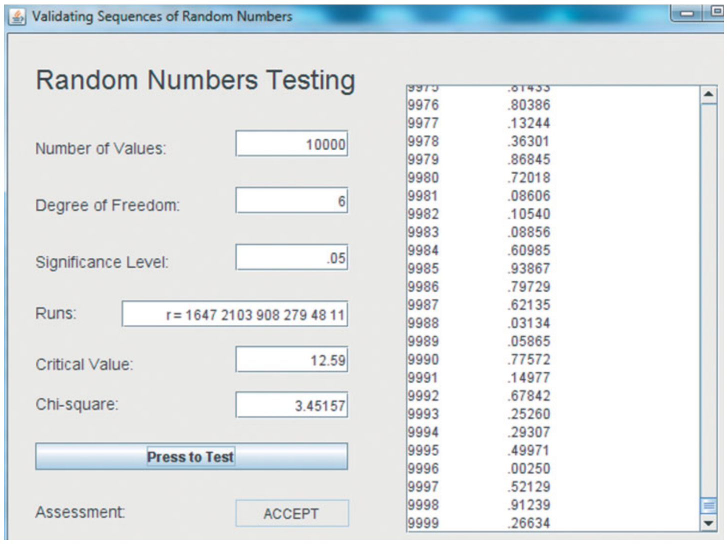

FIGURE A5 | Uniform numbers testing; Ho: random versus Ha: not random for 10000 runs. Ho is NOT rejected.

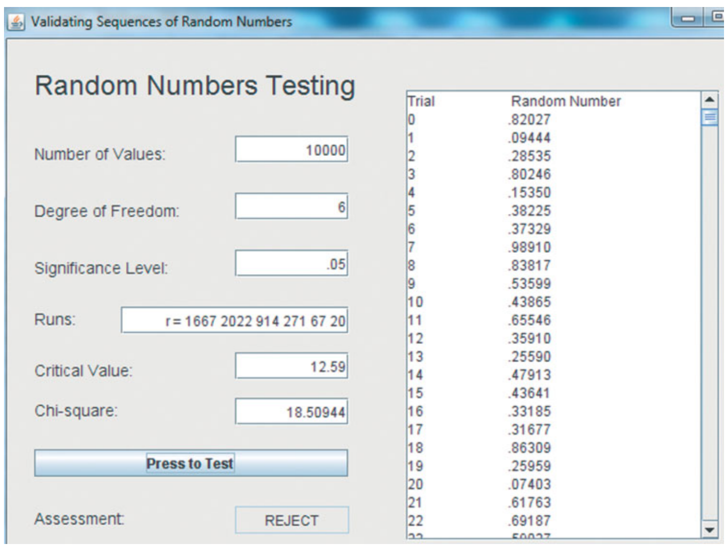

FIGURE A6 | Uniform numbers testing; Ho: random versus Ha: not random for 10,000 runs. Ho is rejected. On the average, one out of 25 cycles of 10,000 = 250,000 simulations will end up rejecting Ho: random.

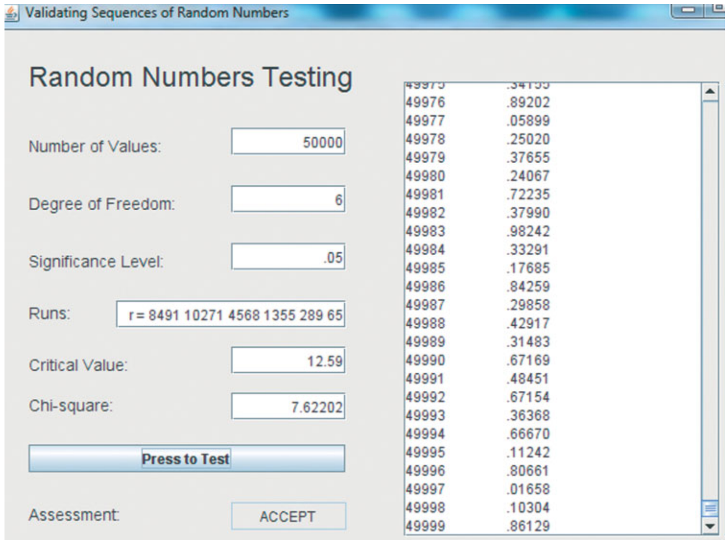

FIGURE A7 | Uniform numbers testing; Ho: random versus Ha: not random for 50,000 runs. Ho is NOT rejected. After 60 cycles × 50K = 3000K = 3,000,000 simulations there is still no reject Ho = random sequence. This may signal a cut-off point of no rejection of random sequence from this point on. Safe threshold may be 50K for JAVA coding uniform random number generator.

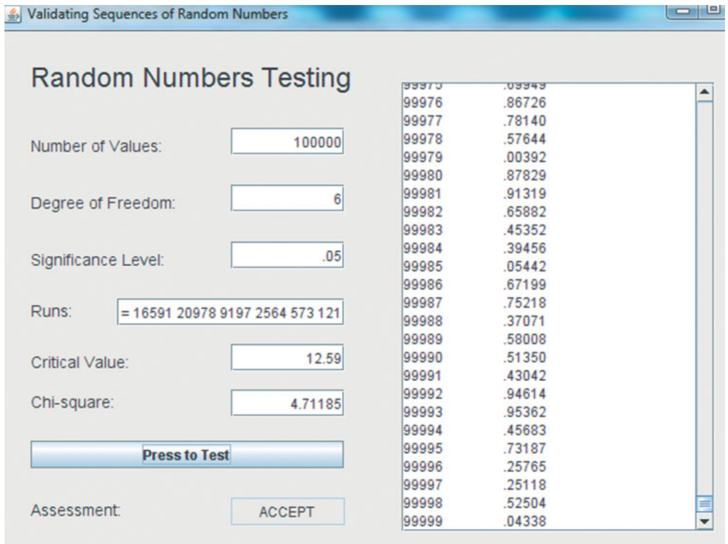

FIGURE A8 | Uniform numbers testing; Ho: random versus Ha: not random for 100,000 runs. Ho is NOT rejected. After 50 cycles × 100K = 5000K = 5,000,000 simulations, there is still no reject Ho = random sequence. This may signal still no rejection of random sequence from the earlier safe threshold: 50K for a JAVA coding uniform random number generator.

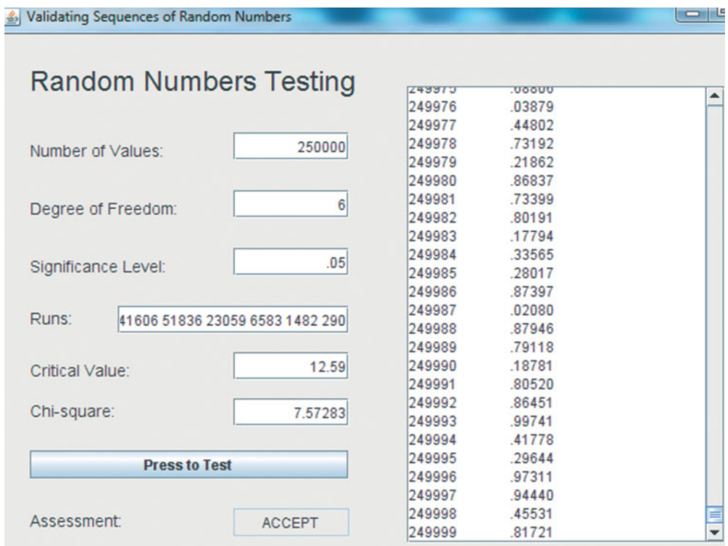

FIGURE A9 | Uniform numbers testing; Ho: random versus Ha: not random for 250,000 runs. Ho is not rejected. After 40 cycles × 250K = 10,000K = 10,000,000 simulations there is still no reject Ho = Random Sequence. This may signal still no rejection of random sequence from the earlier safe threshold: 50K simulations for a JAVA coding uniform random number generator. *Important Note*: In this figure, buttons indicate: No of values = 250,000 (simulation runs), DF = 6 (Section on *Generic Theory*, by Knuth's Technique[10,11]), Significance level (Type-I error) = 5%, Total Runs: 41,606 × **1** + 51,836 × **2** + 23,059 × **3** + 6583 × **4** + 1482 × **5** + 290 × **6.093** (average for >6) = 250,000, where bold numbers from **1** to >**6** are calculated run sizes by Knuth's method. χ^2 calculated = 7.57 < χ^2 critical value = 12.59. Do NOT reject Ho: random sequence.

APPENDIX B

(1) I would be able to provide practical feedback when designing a real world IT project (e.g., software). I could use the simulation to run tests on software before it is implemented on a network. This gives me an advantage over my colleagues, as I am able to safely determine correctness and efficiency of the software. (2) By using M&S, you are able to show your security risk and probability of occurrence within each area of cyber security. By also being the only one at your firm with the M&S experience you have put yourself in a great position for upward mobility or at the very least a project manager or lead with nondestructive testing. (3) If I had M&S background and coworkers did not, I would have the advantage to develop models and run various simulations to aid in my decision-making process. They would not have this capability and their probability of making the correct decisions would be reduced. Not only would my decisions have a greater chance of being correct, but even in the event a bad decision was made; I would have data to support my decisions where my co-workers would not. (4) Working in a Cybersecurity firm and having the knowledge of M&S enables one to experiment with probability of occurrences. This will enable me analyze potential return on investments by product or lifecycle costs. Lastly, being able to test new products and methods will assist me, and corporate management to make the right decisions based on modeling and simulation versus 'gut' feelings.