Computers & Security 89 (2020) 101659

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

A Bayesian network approach for cybersecurity risk assessment implementing and extending the FAIR model

Jiali Wang [a,*], Martin Neil [b], Norman Fenton [b]

[a] School of Electronic Engineering & Computer Science, Queen Mary University of London, Mile End Road, London E1 4NS, United Kingdom
[b] Agena Ltd, Cambridge, United Kingdom

ARTICLE INFO

Article history:
Received 7 March 2019
Revised 26 October 2019
Accepted 1 November 2019
Available online 3 November 2019

Keywords:
Cybersecurity risk assessment
FAIR model
Bayesian networks
Monte Carlo simulation
Risk aggregation
Adversarial risk analysis
Game theory

ABSTRACT

Quantitative risk assessment can play a crucial role in effective decision making about cybersecurity strategies. The *Factor Analysis of Information Risk* (FAIR) is one of the most popular models for quantitative cybersecurity risk assessment. It provides a taxonomic framework to classify cybersecurity risk into a set of quantifiable risk factors and combines this with quantitative algorithms, in the form of a kind of Monte Carlo (MC) simulation combined with statistical approximation techniques, to estimate cybersecurity risk. However, the FAIR algorithms restrict both the type of statistical distributions that can be used and the expandability of the model structure. Moreover, the applied approximation techniques (including using cached data and interpolation methods) introduce inaccuracy into the FAIR model. To address restrictions of the FAIR model, we develop a more flexible alternative approach, which we call FAIR-BN, to implement the FAIR model using Bayesian Networks (BNs). To evaluate the performance of FAIR and FAIR-BN, we use a MC method (FAIR-MC) to implement calculations of the FAIR model without using any of the approximation techniques adopted by FAIR, thus avoiding the corresponding inaccuracy that can be introduced. We compare the empirical results generated by FAIR and FAIR-BN against a large number of samples generated using FAIR-MC. Both FAIR and FAIR-BN provide consistent results compared with FAIR-MC for general cases. However, the FAIR-BN achieves higher accuracy in several cases that cannot be accurately modelled by the FAIR model. Moreover, we demonstrate that FAIR-BN is more flexible and extensible by showing how it can incorporate process-oriented and game-theoretic methods. We call the resulting combined approach "Extended FAIR-BN" (EFBN) and show that it has the potential to provide an integrated solution for cybersecurity risk assessment and related decision making.

1. Introduction

Cybersecurity has become a critical issue for most organizations due to their increasing reliance on IT systems and the increased complexity of the network environment (Cashell et al., 2004). Organizations face a diverse range of cyber-attack risks, which can cause data breach and more serious consequences (IBM, 2018), including forced interruptions in online services, impaired corporate reputation, and ultimately financial losses. To mitigate or even prevent these risks, Cybersecurity Risk Assessment (CRA) is required, since it can support risk managers to prioritize risks, allocate restricted resources to alleviate them, and make further defence decisions (Blakley et al., 2001; Peltier, 2010).

Factor Analysis of Information Risk (FAIR) is a well-known CRA framework (Freund and Jones, 2014; Jones, 2006) and has been widely applied and recognized in academic research (Sendi and Cheriet, 2014; Le et al., 2017; Le, 2019; Park et al., 2018) and industry (ISACA, 2009; The Build Security, 2019). To structure risk analysis, it uses a taxonomy to classify risk (financial loss) into risk factors and represent the relationships between these risk factors as shown in Fig. 1.

FAIR covers more aspects of CRA compared to other prominent CRA frameworks (Wangen et al., 2016). It considers the capability contest between attackers and defenders, vulnerability of information assets, the frequency of successful attacks, and consequent financial losses, which has provided a good foundation for structuring CRA. The FAIR model is a combination of the FAIR taxonomy and statistical techniques and is used to conduct quantitative risk assessment (The Open Group, 2018; The Open Group, 2019). In this paper, we unpick the assumptions and algorithms used in the FAIR model and identify a number of potential serious

* Corresponding author.
E-mail addresses: jiali.wang@qmul.ac.uk (J. Wang), m.neil@qmul.ac.uk (M. Neil), n.fenton@qmul.ac.uk (N. Fenton).

https://doi.org/10.1016/j.cose.2019.101659
0167-4048/

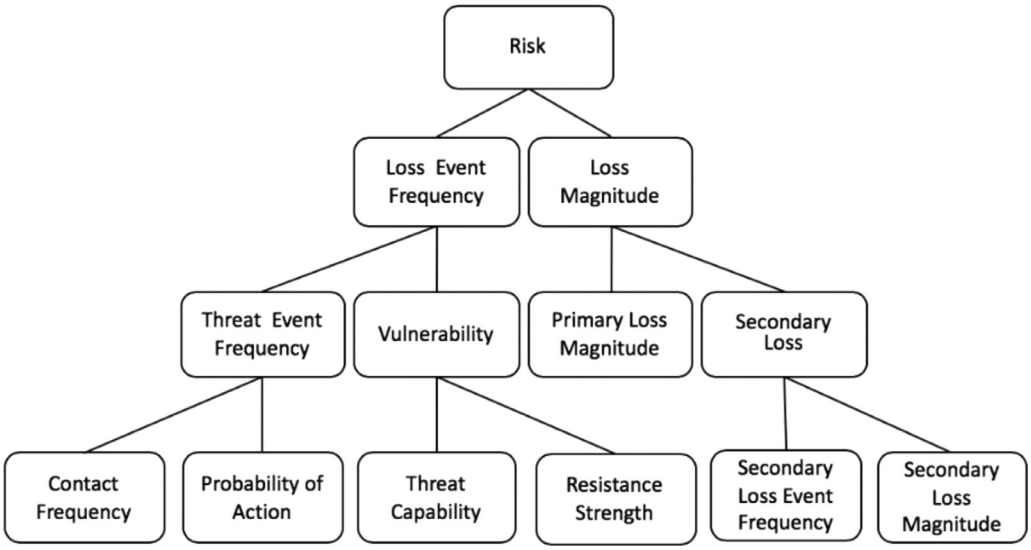


Fig. 1. Taxonomy structure of the FAIR model.

limitations. Firstly, since the FAIR model uses only triangular distributions to simulate input risk factors of the model, alternative statistical distributions (especially long-tailed distributions (Anderson, 2004; Foss et al., 2013)) for input factors may be poorly approximated; this introduces inaccuracy. We provide detailed experimental analysis for this in Section 6. Secondly, it is difficult to extend the model to accommodate other modelling goals and perspectives. To address these limitations, we develop a more flexible alternative approach, which we call FAIR-BN, to implement the FAIR model using Bayesian Networks (BNs) (Nielsen and Jensen, 2009; Fenton and Neil, 2018). FAIR-BN subsumes the existing features of the FAIR model while: (1) allowing a wider set of distributions to represent and process input variables; and (2) supporting both a deeper model of the cyber-attack-defend process and decision making and evaluation.

By employing BNs, we connect the FAIR model with other advanced CRA models to enhance the original model, for example to analyse interactions between attackers and defenders. Interaction between attackers and defenders is a crucial element in CRA, since it influences both risk assessment and decision making about control deployment. However, the related analysis in the FAIR model is simplified and is relatively high-level. BNs have been widely applied in modelling more detailed features of the cyber-attack-defend process, for instance, from the process-oriented perspective, such as attack graphs (Poolsappasit et al., 2012), and from the game-theoretic perspective, such as Adversarial Risk Analysis (ARA) (Banks et al., 2015). In our work, we have incorporated a process-oriented model and a game-theoretic model with the FAIR-BN to provide the integrated risk assessment and management solution. We call them Extended FAIR-BNs (EFBNs).

We evaluate the quantitative accuracy of FAIR and FAIR-BN using results generated by the proposed FAIR-MC and the measurement *J* divergence (Jeffreys, 1946; Lin, 1991). FAIR-MC is a Monte Carlo (MC) simulation based implementation of the FAIR model. We construct FAIR-MC strictly complying with the inference mechanism assumed by the FAIR model. The major difference between FAIR-MC and the FAIR model is in how a core calculation process, called risk aggregation, is performed. The FAIR model uses cached data generated from a kind of MC method combined with statistical approximation techniques, including applying Bounded Metalog Distributions (BMDs) (Keelin, 2016) and an interpolation method to carry out risk aggregation. The application of these approximation techniques introduces inaccuracy into the FAIR model. In comparison, FAIR-MC uses simulation to conduct risk aggregation without using extra approximation techniques and thus avoids introducing the sequential inaccuracy. Moreover, in each test, we generate a much larger number (one million) of samples using FAIR-MC to represent the standard and measure the distance between this standard and results (represented by one thousand samples respectively) generated by the FAIR model and FAIR-BN using *J* divergence. We assume that the smaller *J* divergence the model has against the FAIR-MC, the more accurate the model is.

We empirically compare the results generated by FAIR and FAIR-BN with a focus on accuracy under different statistical scenario assumptions, and in particular 'long tail' assumptions. We use three empirical cases to test if the FAIR model can maintain accuracy in different scenarios where the assumptions differ. We also compare the performance of FAIR-BN against FAIR in all of these cases. Experimental results illustrate that the FAIR-BN and the FAIR model provide consistent results compared with FAIR-MC in general. However, in certain cases, FAIR-BN provides more accurate results, especially in the long-tail case. These evaluation results lay the foundation for confidently implementing and extending the FAIR model using Bayesian Networks.

The contributions of this work are: (1) we provide a detailed and in-depth analysis of the assumptions of the FAIR model, which has hitherto not appeared in the literature, and reveal a number of important limitations; (2) we propose a new approach called FAIR-BN that incorporates the same modelling assumptions used by the FAIR model but also supports wider assumptions and can be more easily extended; (3) we construct a Monte Carlo (MC) simulation (FAIR-MC) without using the approximation techniques that applied by FAIR and introduce *J* divergence (Jeffreys, 1946; Lin, 1991) to perform accuracy evaluation for the FAIR model and FAIR-BNs; (4) we evaluate the performance of the FAIR model and the proposed FAIR-BN and identify cases where the FAIR model produces inaccurate results; (5) we construct EFBNs incorporating the FAIR-BN with a process-oriented model and a game-theoretic model to provide integrated risk assessment and in tandem illustrate how the FAIR-BN can be expanded.

The paper is structured as follows. In Section 2, we introduce related cybersecurity work with a focus on how it might be used in the FAIR-BN approach. In Section 3, we introduce the FAIR model focusing on its taxonomic structure and algorithms. In Section 4, we describe FAIR-BN, i.e. how to faithfully represent the FAIR model using BNs. We describe FAIR-MC and *J* divergence

in Section 5. Experiments evaluating the performance of the FAIR model and the FAIR-BN are provided in Section 6. In Section 7, we provide examples illustrating flexibility and expandability of the FAIR-BN. We discuss pros and cons of the FAIR model, FAIR-BN and FAIR-MC in Section 8 and provide conclusions in Section 9.

2. Related work

FAIR is applied in Le, 2019) to assess loss event frequencies of smart grid cyber threats and is employed by Park et al. (2018) to evaluate threats of Android malware. However, both studies only applied the qualitative framework of FAIR. The FAIR model is applied in Sendi and Cheriet (2014) to analyse risk in cloud computing. In general, these studies fail to provide deeper insight into, or evaluation of, the FAIR model; nor do they suggest how its deficiencies might be addressed. In this paper, we aim to remedy this.

The FAIR model provides a relatively high-level risk assessment framework. To model more detailed features of the cyber-attack-defend process and support decision making and evaluation, we propose extending the FAIR by eliminating its limitation of expandability using our proposed FAIR-BN and by connecting it with other dedicated CRA models. There are two kinds of models that we explore as enhancements of the FAIR-BN, process-oriented risk assessment models and game-theoretic models.

Several process-oriented risk assessment methods for CRA have been developed in recent decades. Typical paradigms include threat trees (LeBlanc and Howard, 2002), attack trees (Schneier, 1999), attack graphs (Jha et al., 2002) and defence trees (Bistarelli et al., 2006). These methods provide graphical notations which illustrate the attacker's goals with possible routes to reach these goals and then help to identify the controls regime required to thwart the attack threat. For instance, the defence tree is an extension of attack trees with added leaves representing controllable countermeasures. However, in general, these studies fail to consider the capability of attackers and the frequency with which that the attack might be successfully executed to endanger financial losses. To alleviate such drawbacks, Bayesian Attack Graphs (BAG) (Poolsappasit et al., 2012; Liu and Man, 2005) and the security graph model (Xie et al., 2010) have been proposed as alternatives. These approaches apply Bayesian probabilistic logic to conduct CRA. However, none of them provides a unified risk aggregation mechanism for producing an interpretable risk evaluation result. In our work, we have implemented a unified risk aggregation mechanism, based on the work of Lin et al. (2014), using BNs and producing financial losses to represent risks.

Game-theoretic methods have also been widely applied to cybersecurity issues (Manshaei et al., 2013; Do et al., 2017; Roy et al., 2010; Wang et al., 2016). Specifically, a game-theoretic model is applied to solve the network defend-attack problem in Lye and Wing (2005). However, a recognized feature of game-theoretic solutions is the lack and asymmetry of information, such as the absence of knowledge about the attacker's strategy domains or payoffs (Nguyen et al., 2009). Several ways to capture the uncertainty in game theory are proposed in Rass et al. (2015), Rass et al. (2017) and Banks et al. (2015). Among them, Banks et al. (2015) is a monograph that describes how to use the perspective of Adversarial Risk Analysis (ARA) to address the uncertainty in game theory. A broad review of applications of ARA in a variety of game contexts (i.e. simultaneous games, sequential games, etc.) is provided in Banks et al. (2015) and extended in Brown et al. (2008) and Alderson et al. (2011). In ARA, the decision problem can be structured and represented by an Influence Diagram (ID), which is a generalization of a BN. For cases with more than one decision maker, Multi-Agent Influence Diagrams (MAIDs) are proposed as an extension of IDs (Koller and Milch, 2003). A game model can hardly guide risk managers for decision making and consequence evaluation individually. Given this, incorporating the game model with unified payoff modelling is a feasible solution but can usually be neglected in game theory studies, especially in the cybersecurity context. In our work, we have explored the combination of a game-theoretic model and the FAIR-BN, which can support decision making about defend deployment in the defend-attack game and predict residual loss posed by cyber-attack within the integrated model.

An attempt to improve the flexibility of FAIR by using BNs was proposed in Le et al. (2017). This work applied a part of the FAIR framework to assess the success frequency of cyber-attack events in smart grids. However, this work does not consider the whole FAIR structure nor use quantitative reasoning. In our work, we present a complete implementation of the FAIR model using BNs and explore directions for improving it by incorporating process-oriented models and game-theoretic models.

Here the application of BNs is focused on their ability to represent probabilistic reasoning (i.e. implementing the FAIR model) and causal reasoning (i.e. the process-oriented model in 7.1) together, derived from subject matter expertise and from the structure of the networked system. However, it is possible, using data alone, to learn the BN graph structure and/or the strengths of statistical associations between variables. In this way, they offer a universal approach to causal and statistical reasoning, complementing the absence of data with expertise and vice versa. Relevant examples include (Gruber and Ben-Gal, 2019; Zhu et al., 2016).

3. Overview of the FAIR model

3.1. FAIR model structure: taxonomy and aggregation

The taxonomy structure of the FAIR model (Freund and Jones, 2014; The Open Group, 2018) was shown in Fig. 1, with the risk classes being modelled. Risk (financial loss) is defined by Loss Event Frequency (LEF) and Loss Magnitude (LM). LEF is defined as the frequency that a threat agent will inflict harm on an information asset within a given timeframe and itself is a function of Threat Event Frequency (TEF) and Vulnerability (V), where the former represents 'the frequency that a threat agent will act against an asset', whilst the latter is defined as 'the probability that an asset will be unable to resist the actions of a threat agent' (Jones, 2006). TEF is the frequency that a threat agent will come into contact with an asset and the probability that a threat agent will act against an asset once contact occurs (referred to as Contact Frequency (CF) and Probability of Action (PoA) respectively). V is the difference between the level of force that a threat agent is capable of applying against an asset (Threat Capability (TC)) and the strength of control (Resistance Strength (RS)). LM is categorized as either a Primary Loss (PL) or Secondary Loss (SL) (these are assumed to be exhaustive and mutually exclusive (The Open Group, 2018)). In the FAIR model, PL represents the direct losses from assets and threats whilst SL represents secondary consequential losses such as negative organizational and external environment after effects. Furthermore, secondary loss is broken down into the Secondary Loss Event Frequency (SLEF) and the Secondary Loss Magnitude (SLM).

The key feature of the FAIR model is that the structure and taxonomy are fixed and cannot be extended, so any differences in assumptions cannot be supported (such as a different, perhaps more detailed way, to model threats and defences).

Fig. 2 shows the FAIR risk aggregation calculations diagrammatically and shows the statistical operations and objects needed to calculate risk using the FAIR taxonomy. The FAIR model makes many, quite reasonable assumptions, but some are implicit. Total losses are calculated by adding primary and secondary losses, each of which is calculated by multiplying loss frequency and loss

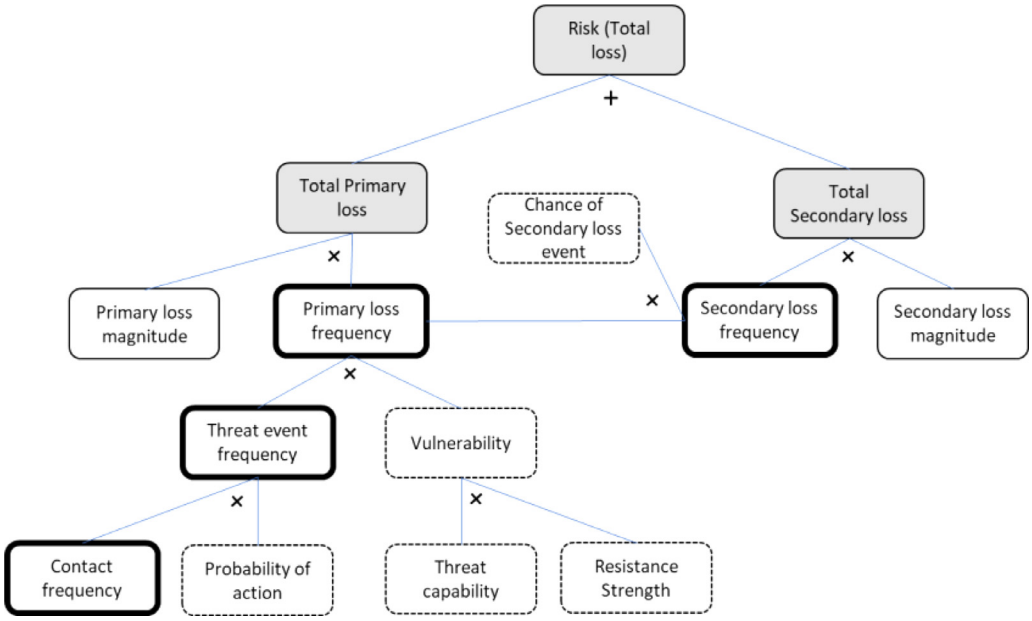


Fig. 2. Risk aggregation structure in the FAIR model (Risks are shown as grey rectangles, frequency measures as boldly outlined rectangles, probability measures as dotted rectangles and financial loss magnitude measures as undashed white rectangles. Operators are shown as (+) or (x) for addition and multiplication).

Table 1
Output and input factors and functions used in the FAIR model.

No.	Output Factor	Input Factor	Function
1	Total Loss (TL): L_T	Primary Total Loss (PTL): L_P Secondary Total Loss (STL): L_S	$L_T = L_P + L_S$;
2	Primary/Secondary Loss (PL/SL): L_P/L_S	Loss Event Frequency (PLEF/SLEF): F_P/F_S Loss Magnitude (PLM/SLM): LM_P/LM_S	$L_P = RA(F_P, LM_P)$; $L_S = RA(F_S, LM_S)$;
3	Mean of Primary Loss Event Frequency (MPLEF): M_{PLEF}	Threat Event Frequency (TEF): F_{TE} Vulnerability (V): P_V	$M_{PLEF} = F_{TE} \times P_V$;
4	Primary Loss Event Frequency (PLEF): F_P	Mean of PLEF (MPLEF): M_{PLEF}	$F_P = \text{Poisson}(\lambda = M_{PLEF})$;
5	Secondary Loss Event Frequency (SLEF): F_S	Primary Loss Event Frequency (PLEF): F_P Chance of Secondary Loss (CSL): P_{SL}	$F_S = Binomial(n = F_P, p = P_{SL})$;
6	Threat Event Frequency (TEF): F_{TE}	Contact Frequency (CF): F_C Probability of Action (PoA): P_A	$F_{TE} = F_C \times P_A$;
7	Vulnerability (V): P_V	Threat Capability (Tcap): P_{TC} Resistance Strength (RS): P_{RS}	$P_V = P(P_{TC} > P_{RS})$

magnitude, but with the caveat that secondary loss events can only occur given that primary loss events have occurred beforehand. In this way, an element of causal conditioning is introduced into the risk aggregation process that is not immediately obvious. Secondary loss frequency is, therefore, a function of the primary loss frequency. If there is zero chance of a secondary loss, then there will no secondary loss events to aggregate. Primary losses are also treated differently from secondary losses in that there are the causal assumptions; frequency of primary losses is calculated from threat event frequency and vulnerability.

3.2. FAIR model algorithms: simulation-based calculation

The FAIR model proposes a series of functions relating variables (risk factors), which statistically or probabilistically represent the functional relationships between a factor and its sub-factors (The Open Group, 2018). We summarize the factors and functions in Table 1.

Analysis proceeds from bottom to top (step 7 to step 1) through the risk aggregation structure using the function declared for each input-output factor combination. The FAIR model is built in Excel and uses an add-in sample generating tool, SIPmath (The Open Group, 2019). In the model, each risk factor is represented as a random variable, from which generated samples are stored in a column of data, which is referred to as a Stochastic Information Packet (SIP). The sample distribution of each factor can either be calculated from its sub-factors or randomly simulated using a triangular distribution specified by the user. Functions listed in Table 1 can be performed on corresponding sample vectors.

Risk assessment through the FAIR model includes two procedures: assessing loss event frequencies (calculating factors 3–7) and aggregating loss magnitudes using assessed frequencies to calculate the total loss (calculating factors 1–2). By simulating samples for input factors and operating these samples following corresponding functions, loss event frequencies can be calculated, which is straightforward.

A key process in FAIR is Risk Aggregation (RA), where the compound sums, L_P and L_S, of n Independently Identically Distributed (IID) loss magnitude random variables, LM_P and LM_S, is computed where n is determined by a value from frequency variables, F_P and F_S, (Lin et al., 2014; Heckman and Meyers, 1983). A Poisson distribution, $Poisson(\lambda)$, is used to model primary loss frequency, F_P, using a mean frequency estimate, M_{PLEF}, following the function $F_P = Poisson(\lambda = M_{PLEF})$. As is shown in (The Open Group, 2019), the FAIR model simplifies the risk aggregation process that could be conducted using Monte Carlo (MC) simulation directly. Instead, the FAIR model uses cached simulation results combined with a sta-

tistical approximation technique to simplify this process for more efficient calculation.

To prepare the cached data, samples of L_P corresponding to F_P and LM_P pairs are simulated, and statistical parameters are derived from the samples and stored. These parameters are then used to construct an approximated quantile distribution function (Bounded Metalog Distribution (BMD)) approximating L_P. After obtaining the BMD of L_P, samples of L_P can be generated from the BMD expression using uniformly distributed random probabilities. We provide details of how the BMD is constructed within the FAIR model in Appendix A and demonstrate how the FAIR model uses BMDs and cached data to produce risk aggregation results in Appendix B.

We have already highlighted the implicit basic causal assumptions about cyber events embedded within the FAIR model, namely that the secondary and primary losses are conditionally dependent, by definition. There is also an implicit statistical assumption in FAIR, namely that triangular distributions are used throughout to model user inputs. However, such distributions might not always be valid or suitable. For instance, an expert may wish to represent their uncertainty about an input parameter using some other statistical distributions or may wish to vary how F_P is calculated, perhaps by including information gained from complementary analysis, such as kill graphs or that derived from adversarial risk analysis. We propose using Bayesian Networks (BNs) as an alternative way to implement, extend the FAIR model and eliminate its restrictions, which is described in Section 4 and 7.

4. Modelling FAIR using Bayesian networks

Bayesian Networks (BNs) are widely used for probabilistic reasoning and have very wide applicability, including enabling statistical reasoning such as machine learning from data (Zhou et al., 2014; Zhou et al., 2013), diagnostic inference and causal reasoning (Fenton and Neil, 2011). In this paper, we use BNs as an alternative to FAIR, in the form of FAIR-BN, and extend FAIR-BN incorporating process-oriented and game-theoretic methods.

A BN is a directed acyclic graph representing a factorization of a joint probability distribution, consisting of nodes representing variables and arcs representing causal or probabilistic relationships (the qualitative part) with probabilistic weights (the quantitative part) sometimes modelled using statistical and deterministic functions. In a BN, each node X_i has an associated probability table, $P((X_i|\, pa(X_i))$, called the Conditional Probability Table (CPT) of X_i given its parent variables, $pa(X_i)$. For a node X_i without parents, the CPT is the marginal probability distribution of X_i, $P(X_i)$. The conditional-independent relationship among variables, represented by the absence of arcs, allows simplification of a BN's joint probability distribution which can be represented by the product of CPTs. Furthermore, the marginal distribution of the child variable can be obtained by marginalizing over its parent variables in the joint distribution (Fenton and Neil, 2018). For example, considering a simple BN consisting of three nodes, in which nodes A and B are parents of node C, and CPTs are $P(A)$, $P(B)$ and $P(C|A, B)$, we can get the joint distribution of this BN from $P(A, B, C) = P(A)\, P(B)P(C|A, B)$ and calculate the marginal distribution of the child node C following $P(C) = \sum_{A,B} P(A, B, C)$.

More generally, the joint distribution of a BN can be calculated following formula (1):

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P((X_i|\, pa(X_i)) \tag{1}$$

This significantly reduces the complexity of inference tasks in BNs. The CPT embodies the probabilistic reasoning mechanism into BNs. More relevant details are carefully explained in Fenton and Neil, 2018). In this paper, we have used AgenaRisk (Agena Ltd. 2019), a commercial BN package, to build BNs and perform calculations. AgenaRisk contains off-the-shelf functions for performing inference on hybrid BNs (those containing both continuous and discrete nodes), influence diagrams and for performing compound sum calculations. FAIR-BN and EFBNs can all be implemented using AgenaRisk.

Table 2
Types of risk aggregation process.

Type	Description	Reasoning Function
RA_1	Risk aggregation based on an individual frequency: F_p/F_s	$L_P = RA_1(F_P, LM_p)$; $L_S = RA_1(F_S, LM_S)$;
RA_2	Risk aggregation based on the joint frequency: $F_{P\&S}$	$L_T = RA_2(F_{P\&S}, LM_P, LM_S)$.

As explained in Section 3, Risk Aggregation (RA) is the core reasoning process of the FAIR model, since all the calculations throughout the model recursively calculate the total loss from the derived loss event frequency and loss magnitude distributions. However, the relationship between the involved factors and the mechanism used when conducting RA is implicit in the FAIR model's assumptions. In this subsection, we reveal the mechanism of the RA process and propose algorithms to implement RA using BNs based on the work in (Lin et al., 2014). Finally, we provide the BN for calculating loss event frequencies which includes calculating output factors 3–7 listed in Table 1. The accuracy of the BN's results is then evaluated. The relevant experimental results are provided in Section 6.

There are two types of risk aggregation (denoted as RA_1 and RA_2) needed in FAIR-BN as shown in Table 2.

To show how RA_1 is implemented using BNs, we introduce the calculation of $L_p = RA_1(F_p,\ LM_p)$ as an example. This calculation is conducted using n-fold convolution (Lin et al., 2014; Heckman and Meyers, 1983). Assuming that, in a given period, a cyber event can happen n times where n is any number between 0 and the upper bound N, and the event has a fixed Loss Magnitude distribution LM_P, the primary loss distribution L_P can be calculated following the n-fold convolution shown by formula (2):

$$L_P = P(0)LP_0 + P(1)LP_1 + P(2)LP_2 + \ldots + P(N)LP_N \tag{2}$$

Here LP_n represents the n-fold distribution of LM_P, with $LP_0 = 0, LP_n = LP_{n-1} + LM_P$ for $n = 1\ to\ N$ and $P(n)$ is the probability of $F_P = n$. This n-fold convolution method, which conducts RA_1 based on probabilistic inference, has been implemented by the compound sum function in AgenaRisk. In Fig. 3, we show a RA_1 result given input distributions for primary loss frequency and magnitude.

The BN shown in Fig. 4 models the relationships among associated variables involved in the risk aggregation process RA_2.

In the RA_2 process, the distribution of Total Loss (TL), L_T, can be calculated by conducting risk aggregation on the joint frequency $F_{P\&S}$ and the corresponding Loss Magnitudes (LM), which is denoted as $L_T = RA_2\ (F_{P\&S},\ LM_P,\ LM_S)$. We, therefore, extend the n-fold convolution represented by formula (2) to that shown in formula (3):

$$L_T = \sum_{n=0}^{N} \left[\sum_{m=0}^{n} P(F_P = n,\ F_S = m) \times (LP_n + LS_m) \right] \tag{3}$$

In formula (3), LP_n represents the n-fold distribution of LM_P with $LP_0 = 0, LP_n = LP_{n-1} + LM_P$ for $n = 1\ to\ N$, whilst LS_n represents the n-fold distribution of LM_S with $LS_0 = 0,\ LS_m = LS_{m-1} + LM_S$ for $m = 1$ to n. The function $P(F_P = n,\ F_S = m)$ is the joint frequency distribution that represents the probability that $F_P = n$ and $F_S = m$. We simplify this expression as $P_{n,m}$. We use the BNs

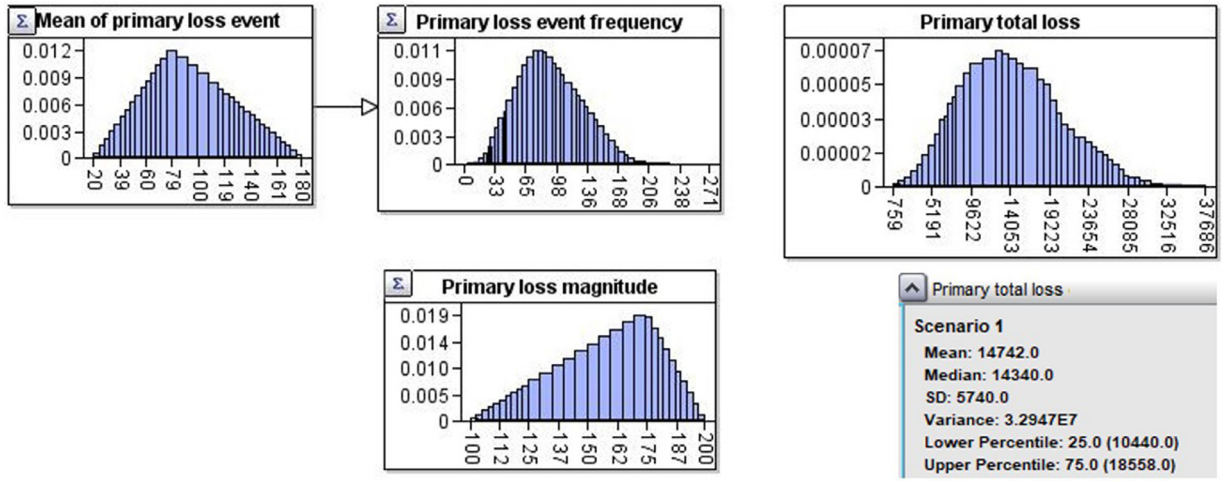


Fig. 3. RA_1result of FAIR-BN with M_{PLEF} following $Triangular(\min = 20,\ ml = 80, \max = 180)$ whilst LM_P followings $Triangular(min = 100, ml = 175,\ max = 200)$.

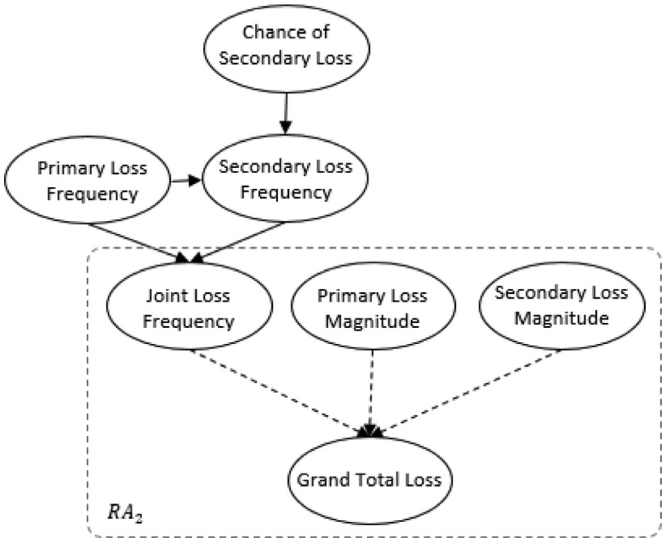


Fig. 4. Risk factors involved in RA_2.

(a), (b) and (c) in Fig. 5 to illustrate the RA_2 process represented by formula (3).

We firstly simplify the BN (c) to BN (e) in Fig. 5, by creating total loss variables $T_{n,\,m}$ which represent the compound results of the associated probability densities, LP_n and LS_m. By doing so, L_T can be calculated by aggregating densities of $T_{n,\,m}$ following the joint frequency distribution. This calculation can be very space inefficient. One solution is to factorize this density aggregation process. A general way of doing so is referred to as the Compound Density Factorization (CDF) method. A CDF method is proposed to calculate RA_1 in Lin et al. (2014). We have extended this 1-Dimension CDF method to a 2-Dimensions CDF method to implement risk aggregation on the joint frequency distribution as the RA_2 process. We use AgenaRisk to implement the related algorithms which are described in Appendix C. An example result showing how RA_2 is calculated is shown in Fig. 6.

Loss event frequency is also modelled in FAIR using some statistical dependencies on threat event frequency and vulnerability variables. These are themselves dependent on contact frequency, probability of action and threat capability and resistance strength respectively. Given BNs can model statistical relationships, they can quite naturally be modelled as shown by the BN in Fig. 7. Additionally, it is possible to extend/replace nodes in this BN to allow us to upgrade a FAIR-BN, incorporating everything FAIR can do, thus providing greater flexibility.

5. Simulation and evaluation using Monte Carlo

5.1. Implementing the FAIR model using Monte Carlo

Monte Carlo (MC) methods are a broad class of computational algorithms that generate numerical results from repeated random sampling (Mahadevan, 1997). In this section, we describe how we use MC simulation methods to implement functions in the FAIR model as listed in Table 1 with a focus on the risk aggregation processes. This series of MC simulations constitute the FAIR-MC. Note that in our FAIR-MC, we do not employ the BMD approximation nor use cached data. This is the most significant difference between the FAIR-BM and the FAIR model.

Firstly, we introduce how we implement the RA_1 process using FAIR-MC. The RA_1 process represents the calculation of primary loss, L_P, using risk aggregation of the corresponding loss event frequency, F_P, and loss magnitude, LM_P. Assuming n samples of F_P have been generated following the specified input distribution (this procedure is straightforward referring to Table 1), for each simulated sample, f_i, of F_P, we simulate LM_P sample f_i times and sum them up to get one sample of L_P. Conducting the same procedure for all samples of F_P, we can get n sample of L_P. The simulation result is a vector of size n, of which each element is represented by formula (4):

$$L_P^i = \sum_{k=0}^{f_i} LM_P^k \tag{4}$$

where $i = 0, \ldots, n$ and LM_P^k represents the kth simulated sample of LM_P following the given distribution.

The method of generating a sample set for secondary loss using FAIR-MC is quite similar. For each sample of the primary loss frequency, f_i, we simulate a sample of the secondary loss frequency f_i' following the Binomial distribution, $Binomial(n = f_i, P = P_{SL}^i)$, where P_{SL}^i represents the ith sample of P_{SL}. Here the occurrence probability of the secondary loss. Then we can generate a sample vector for the secondary loss L_S, with secondary loss magnitude LM_S as formula (5):

$$L_S^i = \sum_{k=0}^{f_i'} LM_S^k \tag{5}$$

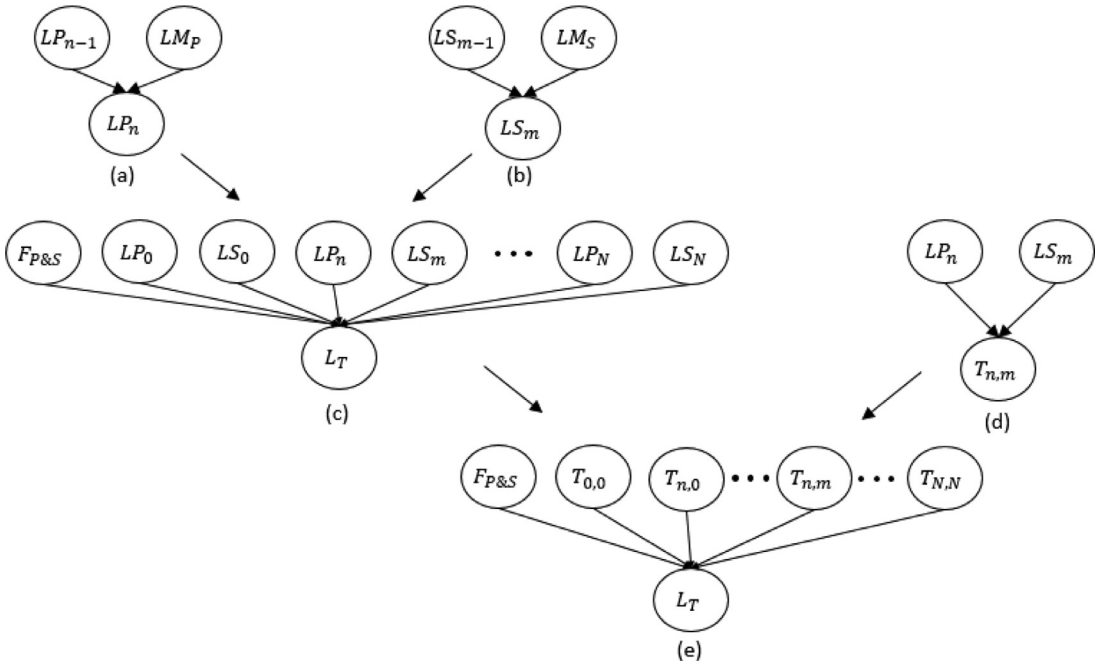


Fig. 5. BNs used to implement the RA_2 risk aggregation process.

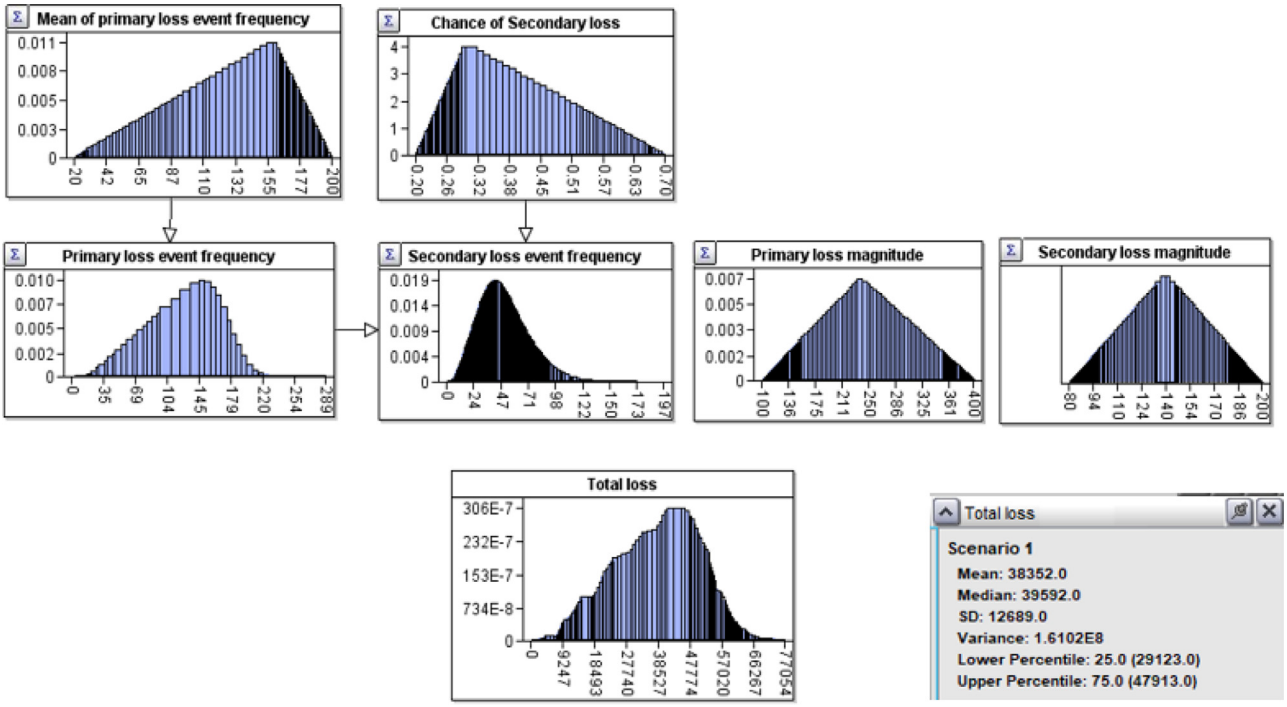


Fig. 6. RA_2 result of FAIR-BN with M_{PLEF} following $Triangular(min = 20, ml = 80, max = 180)$, P_{SL} following $Triangular(min = 0.2, ml = 0.3, max = 0.7)$, LM_P following $Triangular(min = 100, ml = 240, max = 400)$ whilst LM_S following $Triangular(min = 80, ml = 140, max = 200)$.

where $f_i'(i = 0, \ldots, n)$ represents the ith randomly simulated sample of F_S which follows a Binomial distribution, and LM_S^k represents the kth simulated sample of LM_Sfollowing the given distribution.

The sample set of the total loss L_T can be generated based on simulation work above using formula (6):

$$L_T^i = L_P^i + L_S^i \tag{6}$$

Each sample of the total loss L_T^i is calculated by summing the corresponding primary loss L_P^{ij} and secondary loss L_S^{ij}.

Simulating vulnerability, attack capability and, furthermore, the associated primary loss event frequency using FAIR-MC is quite straightforward by generating input samples and operating samples following functions summarized in Table 1.

5.2. Accuracy evaluation

We evaluate the accuracy of the FAIR model by comparing marginal probability distributions produced by the FAIR model

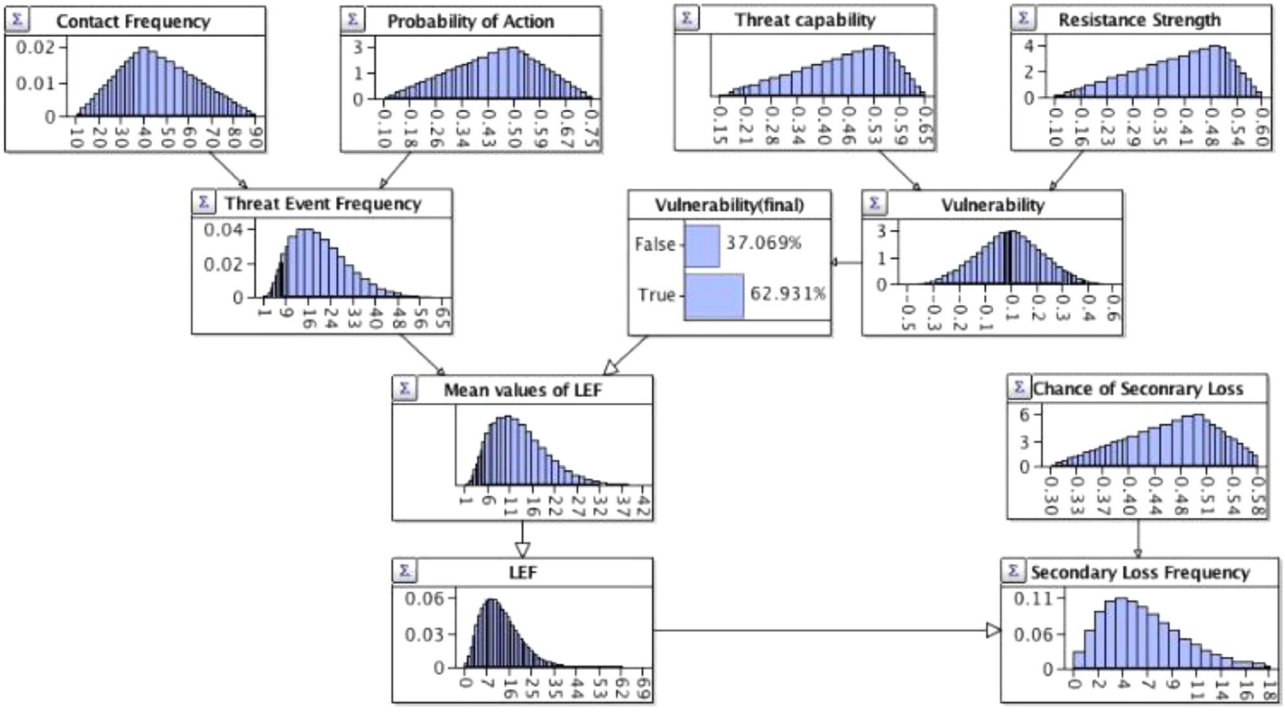


Fig. 7. FAIR-BN for calculating F_P and F_S.

against the marginal probability distributions produced by (a) FAIR-MC simulation and (b) FAIR-BN.

Our aim here is to determine whether the approximation techniques used by FAIR give rise to undesirable inaccuracies and to compare the accuracy of the FAIR model and the corresponding FAIR-BN model.

The accuracy measure we use is based on *K–L* (Kullback–Leibler) divergence, which measures the distance between two distributions, $p(x)$ and $q(x)$ shown by formula (7), (Kullback and Leibler, 1951):

$$\mathrm{K}(p \parallel q) = \int p(x) ln \frac{p(x)}{q(x)} dx \tag{7}$$

Since $K(p\|q)$ is not a symmetric measurement, instead we use a symmetric divergence measure referred to as J divergence shown by formula (8) (Jeffreys, 1946; Lin, 1991):

$$\mathrm{J}(p, q) = K(p \parallel q) + K(q \parallel p) \tag{8}$$

For each function listed in Table 1, we use FAIR-MC to simulate the output factor using a large number of samples (one million). Then, we apply J divergence to measure the distance between the sample distribution calculated by FAIR-MC against results generated by the FAIR model and the FAIR-BN for each output risk factor. The smaller J divergence the model has against the FAIR-MC, the more accurate the model is.

6. Experimental analysis

Our experiments are designed to test the performance of the FAIR model and the FAIR-BN in diverse scenarios. We use one million samples generated by FAIR-MC as the standard to evaluate the results of the FAIR model and FAIR-BN. In Section 6.1 we evaluate whether FAIR-BN can produce consistent results when it complies strictly with the calculation assumptions encoded within the FAIR model. These rules include using only triangular distributions as inputs and the use of functions summarized in Table 1. These evaluation results lay the foundation for confidently implementing and extending the FAIR model using BNs.

In Section 6.2 we consider more realistic scenarios that do not adhere to the strict assumptions underlying the FAIR model. In practical cases, the input data would be much more diverse and complicated. For example, there could be a burst in the frequency of an information asset being attacked in a timeframe. An indication of this could be the existence of Advanced Persistent Threat (APT) (Tankard, 2011). APT can make the targeted information asset dormant under attacks for a long time period. For this reason, using right-long-tailed distributions (Foss et al., 2013; Anderson, 2004), that recognize the probability of extremely large frequencies, to represent the frequency of cyber events is realistic. FAIR's triangular distributions would be a poor approximation in such scenarios, hence introducing inaccuracy. Poor approximations of the data generation process underlying the Loss Event Frequency (LEF) and Loss Magnitude (LM) can directly influence the output of the model (the ultimate assessment of financial losses posed by cyber events). For this reason, we focus our experiments on the RA_1 process and have considered two practical scenarios when LEF follows long-tailed distributions and when LEF is small. Likewise, given the FAIR model employs cached data and approximation techniques to simplify the calculation, its resulting accuracy may be more strongly impaired when LEF takes fixed values that fall between cached values. We have evaluated performance of the FAIR model and the FAIR-BN under these three cases in Section 6.2.

The results of the FAIR model are generated using the Open Fair™ Risk Analysis Tool (The Open Group, 2019), which is built using Excel. Its method of calculation is described in The Open Group, 2018. We have carefully analysed this and have provided more detailed explanation in Section 3, Appendix A and Appendix B. We have used AgenaRisk (Agena Ltd. 2019), a commercial BN package, to build FAIR-BNs and perform calculations. We also have implemented the RA_2 process by developing a program using the AgenaRisk Java API. Related theory and algorithm details are provided in Section 4 and Appendix C.

We use Matlab (MATLAB, 2018) to generate samples following the Monte Carlo (MC) method for each test and call them the results of FAIR-MC. One million MC samples are used in each test

Table 3
Results comparison of L_P distributions with inputs following triangular distributions.

Test	MPLEF			Mean			Variance			99th			J(FAIR, FAIR-MC)	J(FAIR-BN, FAIR-MC)
	Min	Mid	max	FAIR	FAIR-BN	FAIR-MC	FAIR	FAIR-BN	FAIR-MC	FAIR	FAIR-BN	FAIR-MC		
1	0	20	90	5877	5870	5804	9.8E+06	1.1E+07	1.0E+07	13,795	14,339	13,800	0.0174	0.0161
2	0	230	300	28,504	27,857	27,964	1.0E+08	1.2E+08	1.1E+08	46,787	49,883	46,792	0.0332	0.0323
3	20	80	180	14,946	14,730	14,778	2.8E+07	3.3E+07	3.0E+07	27,641	28,643	27,537	0.0230	0.0195
4	60	250	400	37,986	37,458	37,473	1.2E+08	1.4E+08	1.3E+08	60,939	63,143	61,108	0.0239	0.0354
5	20	250	630	48,359	47,237	47,517	3.8E+08	3.8E+08	4.0E+08	92,552	93,069	93,838	0.0194	0.0189
6	15	30	250	15,844	15,686	15,560	7.2E+07	7.1E+07	7.5E+07	36,490	36,500	36,782	0.0168	0.0160
7	15	30	540	31,587	31,168	30,890	3.6E+08	3.9E+08	3.8E+08	77,924	78,953	78,070	0.0312	0.0225
												Average:	0.0236	0.0230

Table 4
Results comparison of L_T distributions.

Test	Description	MPLEF			Mean			Variance			99th			J(FAIR, FAIR-MC)	J(FAIR-BN, FAIR-MC)
		Min	Mid	max	FAIR	FAIR-BN	FAIR-MC	FAIR	FAIR-BN	FAIR-MC	FAIR	FAIR-BN	FAIR-MC		
1	Include 0	0	200	500	7.2E+04	7.1E+04	7.1E+04	9.4E+08	1.0E+09	1.0E+09	1.4E+05	1.5E+05	1.4E+05	0.0362	0.0119
2	Long tail	50	200	1000	1.3E+05	1.3E+05	1.3E+05	3.9E+09	4.2E+09	4.1E+09	2.8E+05	2.8E+05	2.8E+05	0.0237	0.0122
3	Left s kew	20	80	200	3.1E+04	3.1E+04	3.0E+04	1.3E+08	1.5E+08	1.4E+08	5.8E+04	5.9E+04	5.9E+04	0.0320	0.0214
4	Right skew	20	160	200	3.9E+04	3.8E+04	3.8E+04	1.4E+08	1.6E+08	1.5E+08	6.2E+04	6.3E+04	6.3E+04	0.0264	0.0202
5	0 and long tail	0	200	1000	1.2E+05	1.2E+05	1.2E+05	4.1E+09	4.4E+09	4.4E+09	2.8E+05	2.8E+05	2.8E+05	0.0237	0.0145
													Average:	0.0284	0.0160

to reflect the distribution of the output factor. In all of the tests, we use one thousand samples generated by FAIR-BN and the FAIR model respectively to represent the results from the two models. We provide mean, variance, and 99th quantile statistics for the risk aggregation results generated by FAIR, the FAIR-BN and FAIR-MC as a basis for comprehensive comparison. Furthermore, we use *J*divergence to measure distance between FAIR-MC results and results generated by the FAIR model and the FAIR-BN for comparing accuracy of the models.

6.1. Experimental tests complying with assumptions of the FAIR model

6.1.1. Experimental tests of risk aggregation processes

With the assumptions of the FAIR model, LM_P follows a triangular distribution. We use *Triangular* ($\min = 100,\ ml = 175,\ \max = 200$), whose parameters *min, max* and *ml* represent lower bound, upper bound and most likely value, to simulate LM_P in the RA_1 process. In tandem with this, we change parameters of the M_{PLEF} distribution across test cases and furthermore set the distribution of F_P by *Poisson* ($\lambda = M_{PLEF}$) to force diverse shape combinations of F_P and LM_P.

These three methods generate consistent results for L_P. In our seven tests, the average value of *J(FAIR, FAIR-MC)* is 0.0236 while the average value of *J(FAIR-BN, FAIR-MC)* is 0.0230. This shows that, given the same input parameters for M_{PLEF} and LM_P, the $L_P$outputs generated by the FAIR model and the FAIR-BN models are consistent with distributions generated by FAIR-MC. More detailed statistics for the seven experimental tests are shown in Table 3. We also use Euclidean distance (Danielsson, 1980) to measure the distance between FAIR-MC against FAIR and FAIR-BN. We use *Eu(FAIR, FAIR-MC)* and *Eu(FAIR-BN, FAIR-MC)* to represent Euclidean distance between FAIR and FAIR-BN against FAIR-MC respectively. In the seven tests recorded in Table 3, the average *Eu(FAIR, FAIR-MC)* vs *Eu(FAIR-BN, FAIR-MC)* is 0.0283 vs 0.0232. This result confirms that the three models provide consistent results in these seven tests. More detailed results of the Euclidean measurement are given in Appendix E (Table 12).

Next, we experiment on the RA_2 process, considering $L_T = RA_2(F_{P\&S},\ LM_P,\ LM_S)$. To keep inputs consistent with the FAIR model, our experiments on the RA_2 process follow the calculations shown in Fig. 8, where boldly outlined nodes represent input variables that are specified using triangular distributions in the FAIR model.

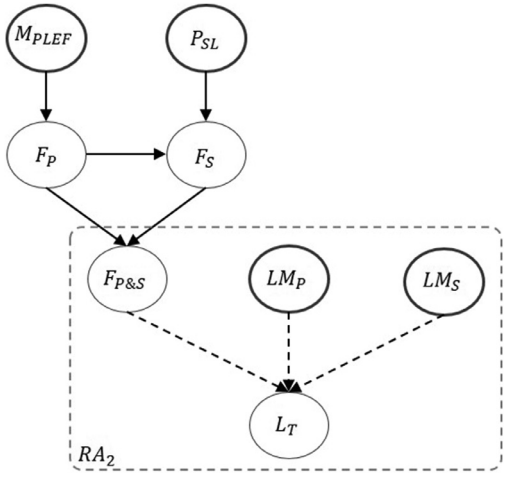


Fig. 8. Related variables in the RA_2 process.

In our experimental tests, LM_P follows *Triangular*($\min = 100,\ ml = 200,\ \max = 400$), LM_S follows *Triangular*($\min = 80,\ ml = 140,\ \max = 200$) and P_{SL} follows *Triangular*($\min = 0.2,\ ml = 0.3,\ \max = 0.7$). Five typical shapes are assigned to M_{PLEF} to construct test cases. We show experimental results of the RA_2 process in Table 4.

Again, the FAIR and the FAIR-BN models generate consistent L_T distributions compared with the FAIR-MC results. The average value of *J(FAIR-BN, FAIR-MC)* is 0.0160 while the average value of *J(FAIR, FAIR-MC)* is 0.0284. This shows FAIR-BN and FAIR generate consistent results when implementing the RA_2 process and the FAIR-BN model generates slightly more accurate results. We also use Euclidean distance to measure the distance between FAIR-MC against FAIR and FAIR-BN for the confirmation. The average *Eu(FAIR, FAIR-MC)* vs *Eu(FAIR-BN, FAIR-MC)* is 0.0378 vs 0.0178, which confirms that the three models provide consistent results in

Table 5
Results comparison of L_P distributions with F_P following long-tailed distributions.

Test	Input Distributions		Mean			Variance			99th			J(FAIR, FAIR-MC)	J(FAIR-BN, FAIR-MC)
	PLEF	PLM	FAIR	FAIR-BN	FAIR-MC	FAIR	FAIR-BN	FAIR-MC	FAIR	FAIR-BN	FAIR-MC		
1	Weibull	LogNormal	1.4E+04	1.4E+04	1.4E+04	4.3E+07	9.8E+07	9.0E+07	29,705	43,254	42,892	0.7683	0.0074
2	Log Normal	LogNormal	3.7E+03	3.5E+03	3.5E+03	2.2E+06	4.3E+06	4.0E+06	7305	10,034	10,218	0.5412	0.0161
3	Gamma	LogNormal	6.4E+03	6.2E+03	6.1E+03	1.0E+07	2.1E+07	2.0E+07	14,275	20,927	20,663	0.5102	0.0073
											Average:	0.6066	0.0103

these five tests. More detailed results of Euclidean measurement are given in Appendix E (Table 13).

6.1.2. Experimental tests of subsidiary risk factors in the FAIR model

In addition to the risk aggregation processes RA_1 and RA_2, there are four other functions applied in the FAIR model as listed in Table 1. We have implemented these in FAIR-BN and FAIR-MC:

1. The Mean of Primary Loss Event Frequency (MPLEF) is calculated from the Threat Event Frequency (TEF) and Vulnerability (V): $M_{PLEF} = F_{TE} \times P_V$. For this, the average value of *J(FAIR-BN, FAIR-MC)* is 0.0069 and the average value of *J(FAIR, FAIR-MC)* is 0.0310.
2. The Primary Loss Event Frequency (PLEF) is derived from MPLEF following $F_P = Poisson\ (\lambda = M_{PLEF})$: here the average value of *J(FAIR, FAIR-MC)* is 0.0170 and the average value of *J(FAIR-BN, FAIR-MC)* is 0.0059.
3. The Secondary Loss Event Frequency (SLEF) is computed from PLEF and Chance of Secondary Loss (CSL) following $F_S = Binomial\ (n = F_P, P = P_{SL})$ and this produces an average value of *J(FAIR, FAIR-MC)* = 0.0213 and the average value of *J(FAIR-BN, FAIR-MC)* = 0.0053.
4. The outputs of Vulnerability, which are derived from Threat Capability (Tcap) and Resistance Strength (RS) following $P_V = P(P_{TC} > P_{RS})$ are probabilities. The FAIR model and FAIR-BN have similar performance.

Experimental results above show that, in calculating LEF and its sub-factors, both the FAIR model and FAIR-BN provide consistent results compared with FAIR-MC. We provide more detailed results of these experimental tests in Appendix D (Tables 8, 9, 10 and11).

6.2. Experimental tests of other practical scenarios

Here we evaluate the performance of the FAIR model and the FAIR-BN in the RA_1 process under two scenarios where LEF follows long tailed distributions and where LEF is small. Also, given the FAIR model employs cached data and statistical techniques in simplifying the calculation, we also evaluate performance in the RA_1 process where LEF has several fixed values, i.e. where poor approximation might be most evident. We focus the experiments on the RA_1 process in this subsection since it is the core calculation in the FAIR model and can directly influence the output of the model (the ultimate assessment of financial losses posed by cyber events).

6.2.1. LEF follows long-tailed distributions

We use three right-long-tailed distributions (which have the possibility of extremely large values) to represent the LEF:

- Weibull distribution (shape = 1.5, scale = 100)
- Log Normal distribution (mean = 3, standard deviation = 0.5)
- Gamma distribution (alpha = 2, beta = 20)

Since these are continuous distributions, to keep their features and model frequencies, we have used each of them as the parameter λ for a Poisson distribution to construct the discrete integer distributions for the corresponding LEF in our test. LM in these tests follows a Log Normal distribution (mean = 5, standard deviation = 0.25). Results generated using the FAIR model, FAIR-BN and FAIR-MC for $L_P = RA_1(F_P,\ LM_P)$ are recorded in Table 5. We compare distributions of primary losses, L_P, generated by these three models in Fig. 9. The average *J(FAIR-BN, FAIR-MC)* is 0.0103 in these three test scenarios. This is consistent with *J(FAIR-BN, FAIR-MC)* in the general cases shown in Table 3. However, the average *J(FAIR, FAIR-MC)* is 0.6066, which is significantly larger than the average *J(FAIR-BN, FAIR-MC)*. The experimental results demonstrate that the FAIR model losses accuracy when dealing with long tailed distributions, while FAIR-BN provides more accurate results that are consistent with results generated by FAIR-MC. This is illustrated intuitively in Fig. 9. We also use Euclidean distance as an alternative measurement in this test group, and the results can lead to the consistent conclusion. More detailed results of this are given in Appendix E (Table 14).

6.2.2. LEF and LM using other statistical distributions

In addition to the long tail distribution scenario, there are other situations that may require different distributions rather than those assumed by FAIR. For example, FAIR uses a Poisson distribution, with an input triangular distribution, to simulate the LEF for the further risk aggregation. In practice, the Binomial distribution is better suited to model frequency distributions with low values of n and higher values for p (the Poisson is the limit version of the Binomial where n is large and the probability of success, p, is small).

We conduct four tests (whose statistical and graphical results are shown in Table 6 and Fig. 10, respectively). To simulate LEF, we use a Binomial distribution (number of trials = 50, probability of success = 0.2) in tests 1 - 2 and a Triangular distribution (min = 10, ml = 60, max = 100) in tests 3 - 4. For LM, we use a Triangular distribution (min = 100, ml = 175, max = 200) in test 1, a Log Normal distribution (mean = 5, standard deviation = 0.25) in tests 2 - 3 and a Gamma distribution (alpha = 8, beta = 30) in test 4.

The results show that FAIR is less accurate than FAIR-BN and does not even achieve the accuracy that FAIR has in general cases, that we analysed in Section 6.1. The average *J(FAIR-BN, FAIR-MC)* in this test group is 0.0354 which is consistent with the general cases shown in Table 3. However, the average *J(FAIR, FAIR-MC)* is 0.2227, which is much larger than the average *J(FAIR-BN, FAIR-MC)*. The statistics shown in Table 6 and distribution comparisons shown in Fig. 10, demonstrate the insufficiency of FAIR in the RA_1 process when it approximates distributions of input variables using triangular distributions.

6.2.3. LEF with fixed values

Given the FAIR model applies approximation techniques to implement risk aggregation, we apply seven tests involving loss event frequencies that are of fixed values rather than distributions, since it is here that poor approximation might be most evident.

As shown in Table 7, mean, variance and 99th quantile values of results generated by FAIR, the FAIR-BN and FAIR-MC models are consistent with each other across all tests. The average of J divergence between FAIR-BN and FAIR-MC is lower than that between

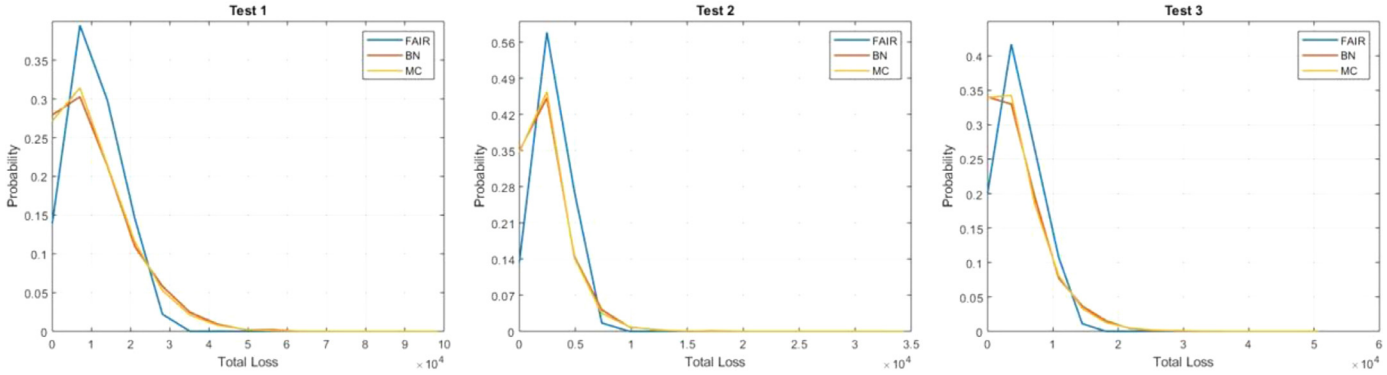


Fig. 9. Results comparison of L_P distributions with F_P following long-tailed distributions.

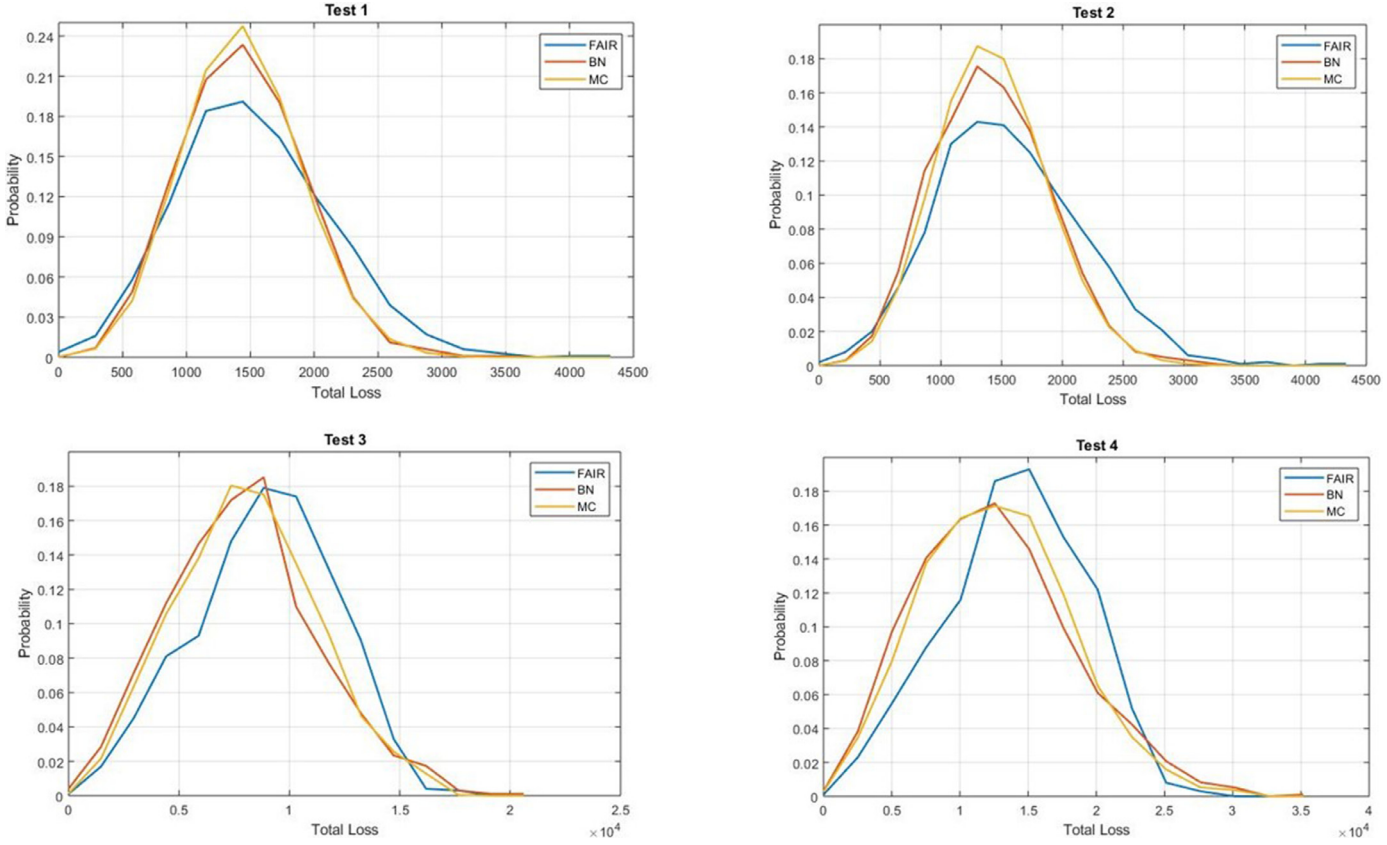


Fig. 10. Results comparison of L_P distributions with F_P and LM_P following other distributions.

FAIR and FAIR-MC (0.0183 vs 0.0768), leading to the conclusion that the FAIR-BN model is more accurate in this scenario.

6.3. Summary conclusions from experiments

We can conclude that both the FAIR and FAIR-BN models can provide consistent results compared with the FAIR-MC standard. However, given that FAIR focuses on simulation efficiency, approximates input variables using triangular distributions and uses cached data and the interpolation method, the model shows insufficiency in dealing with cases when LEFs follow long tailed distributions, LEF and/or LM follow other distributions (rather than triangular distributions) and LEF are of fixed values. In these three scenarios, the FAIR model shows inaccuracy when conducting the RA_1 process. In comparison, the FAIR-BN model provides highly accurate results across all the experimental tests.

7. The extended FAIR-BN models

A wider range of distributions can be applied to represent input factors in the FAIR-BN including for risk aggregation processes and the assessment of loss event frequencies. Here we show an example to demonstrate that the FAIR-BN model can accommodate different distributions for the loss event frequency and loss magnitude factors used in risk aggregation (and could easily do so elsewhere). Fig. 11 shows a FAIR-BN model result achieved by computing RA_1 with Truncated Normal (TNormal) distributions (Robert, 1995) being assigned to primary loss event frequency, F_P, and the corresponding loss magnitude, LM_P, rather than a Poisson distribution and a triangular distribution used in the FAIR model. We use TNormal distributions with 0 as their lower bounds to represent PLEF and PLM are not negative. Other rational distributions can be applied as well.

Table 6
Results comparison of L_P distributions with F_P and LM_P following other distributions.

Test	Input Distributions		Mean			Variance			99th			J(FAIR, FAIR-MC)	J(FAIR-BN, FAIR-MC)
	PLEF	PLM	FAIR	FAIR-BN	FAIR-MC	FAIR	FAIR-BN	FAIR-MC	FAIR	FAIR-BN	FAIR-MC		
1	Binomial	Triangular	1.7E+03	1.6E+03	1.6E+03	3.6E+05	2.2E+05	2.1E+05	3215	2798	2705	0.1748	0.0138
2	Binomial	Log Normal	1.7E+03	1.5E+03	1.5E+03	3.6E+05	2.3E+05	2.1E+05	3237	2733	2656	0.2357	0.0176
3	Triangular	Log Normal	9.6E+03	8.6E+03	8.8E+03	9.8E+06	9.9E+06	9.9E+06	15,811	16,720	16,437	0.2240	0.0570
4	Triangular	Gamma	1.5E+04	1.4E+04	1.4E+04	2.5E+07	3.1E+07	2.8E+07	25,186	28,721	27,864	0.2561	0.0531
											Average:	0.2227	0.0354

Table 7
Results comparison of L_P distributions with F_P is of fixed values.

Test	LEF	Mean			Variance			99th			J(FAIR, FAIR-MC)	J(FAIR-BN, FAIR-MC)
		FAIR	FAIR-BN	FAIR-MC	FAIR	FAIR-BN	FAIR-MC	FAIR	FAIR-BN	FAIR-MC		
1	60	9.5E+03	9.5E+03	9.5E+03	2.7E+04	2.8E+04	2.8E+04	9.9E+03	9.9E+03	9.9E+03	0.0996	0.0193
2	120	1.9E+04	1.9E+04	1.9E+04	5.4E+04	5.6E+04	5.6E+04	2.0E+04	2.0E+04	2.0E+04	0.0805	0.0163
3	175	2.8E+04	2.8E+04	2.8E+04	7.9E+04	8.0E+04	8.0E+04	2.8E+04	2.8E+04	2.8E+04	0.0626	0.0159
4	230	3.6E+04	3.6E+04	3.6E+04	1.0E+05	1.1E+05	1.1E+05	3.7E+04	3.7E+04	3.7E+04	0.0719	0.0217
5	310	4.9E+04	4.9E+04	4.9E+04	1.4E+05	1.5E+05	1.6E+05	5.0E+04	5.0E+04	5.0E+04	0.0666	0.0179
6	390	6.2E+04	6.2E+04	6.2E+04	1.7E+05	1.8E+05	1.9E+05	6.3E+04	6.3E+04	6.3E+04	0.0875	0.0215
7	630	1.0E+05	1.0E+05	1.0E+05	2.8E+05	2.9E+05	2.9E+05	1.0E+05	1.0E+05	1.0E+05	0.0686	0.0156
										Average:	0.0768	0.0183

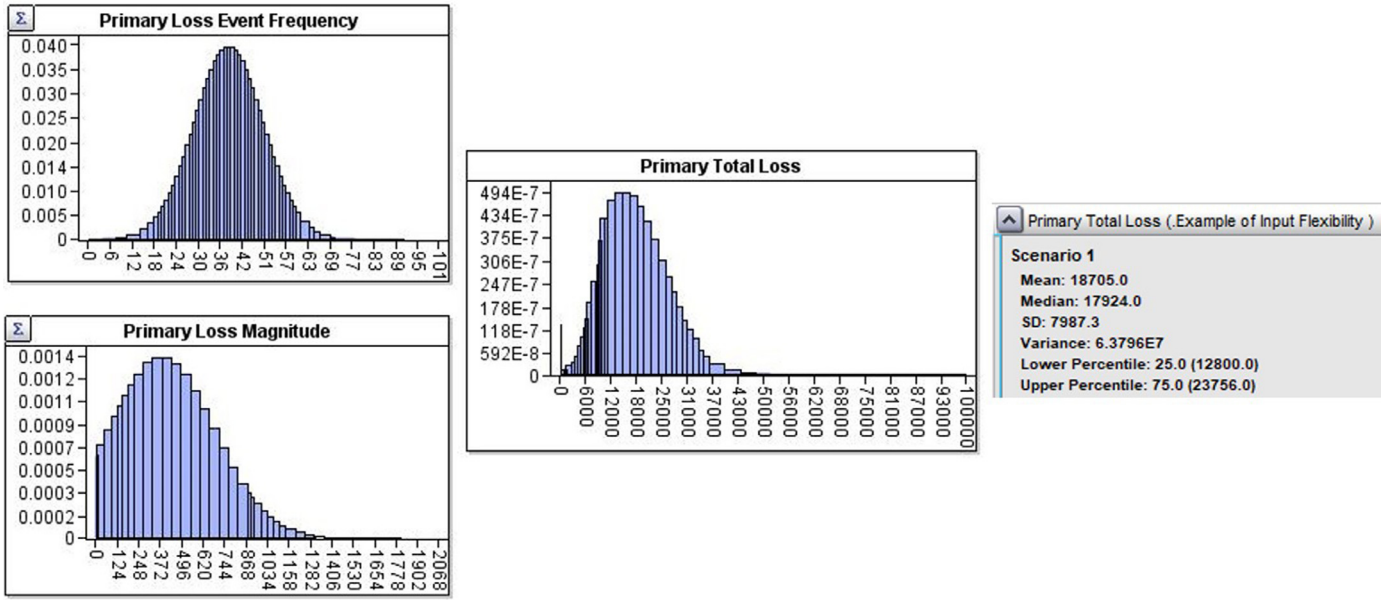


Fig. 11. A RA_1 result of FAIR-BN with inputs following TNormal distributions. F_P follows $TNormal(\mu = 40,\ \sigma^2 = 100,\ LowerBound = 0,\ UpperBound = 200)$ whilst LM_P follows $TNormal(\mu = 400,\ \sigma^2 = 100,000,\ LowerBound = 0,\ UpperBound = 10,000)$.

7.1. Extending the FAIR-BN using a process-oriented model

In addition to providing more flexibility when modelling input distributions and providing more accurate results, more importantly, the FAIR-BN can be easily extended to model the causal processes that represent the interactions between cyber attackers and defenders. The FAIR-BN model can, therefore, be customized to model these factors directly, as cause-effect relationships with associated probabilities. Here we show how we might integrate a simple process-oriented model into the FAIR-BN, replacing the calculation of the vulnerability variable in FAIR by a richer causal structure. We show this model in Fig. 12.

In this model, an information asset is assumed to have three vulnerable aspects (vulnerability *X, Y, Z*) that can be attacked by a threat agent, whilst the threat agent has the capability to attack and exploit each of the vulnerabilities. Controls A and B in the example model can be deployed to reduce the vulnerabilities for one or more vulnerable aspects. Each control is characterised by Operational Effectiveness (OE) which is its probability of reducing vulnerability (i.e. controls are not perfect). The OE of a control is determined by two factors: the extent of deployment and design effectiveness. The output of the control scenario is the vulnerability which represents the probability that the threat agent delivers an attack to the asset successfully. The conditioning logic connecting the variables could be modelled using simple Boolean "AND" and "OR" relationships and CPTs could be elicited from expert knowledge. The probabilities used in this model are an example, which will not influence the reasoning mechanism which we have described in Section 4. Similarly, other process-oriented risk assessment models, such as the kill chain model (Hutchins et al., 2011)

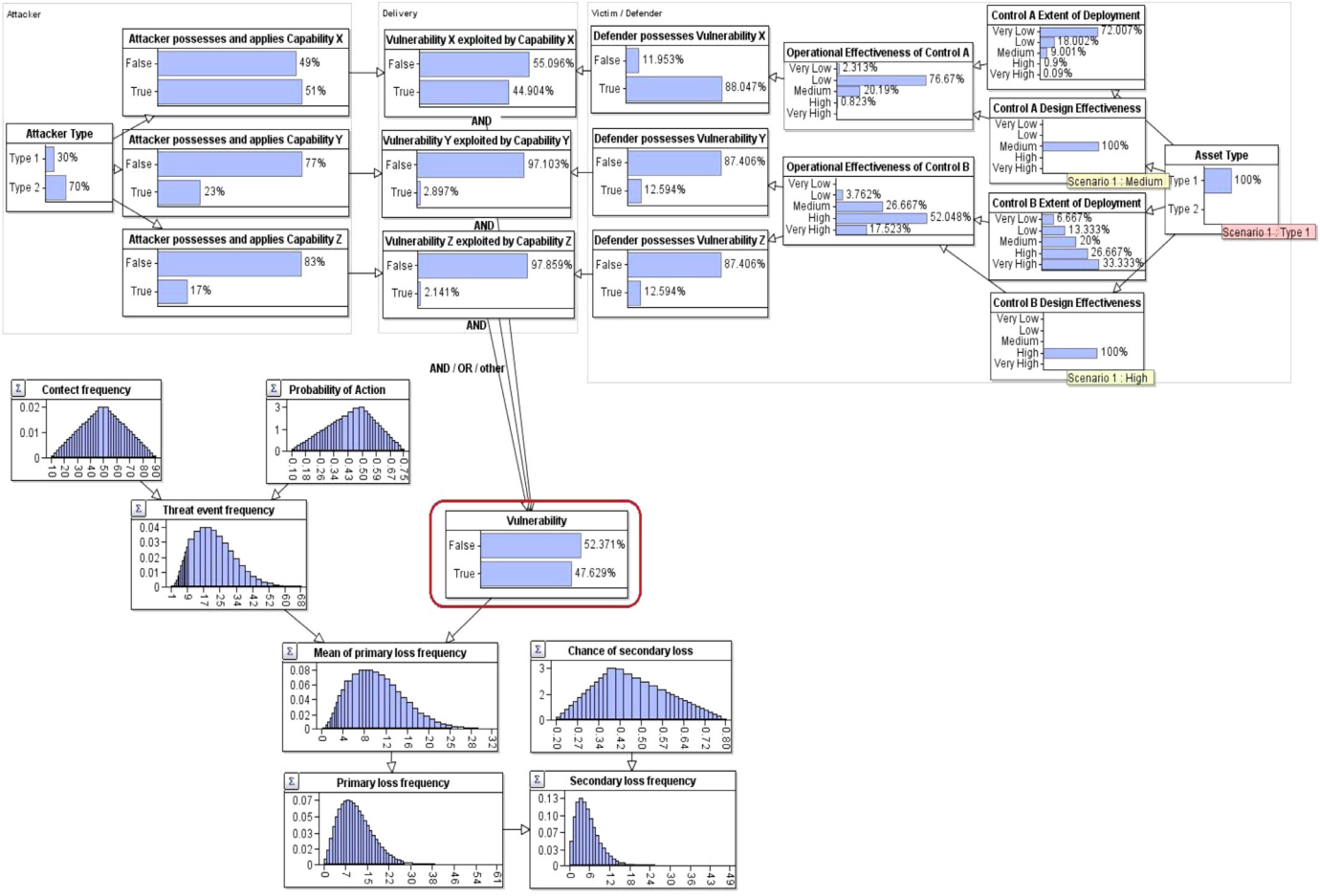


Fig. 12. FAIR-BN extended by a process-oriented model.

and attack graphs (Poolsappasit et al., 2012; Liu and Man, 2005) can be combined with the FAIR-BN for more advanced risk assessment.

7.2. Extending the FAIR-BN using game theory (Adversarial Risk Analysis)

In the FAIR model, the vulnerability of an information asset is determined by a contest between attackers and the defender. The classical game theory finds the Nash equilibrium for all players simultaneously and therefore provides an optimum solution to this contest. However, relatively new methods such as Adversarial Risk Analysis (ARA) (Rios Insua et al., 2009) provide an alternative solution whereby the decision problem is analysed from the view of a specific decision maker (attacker or defender). For example, from the defender's point of view, a model considers the likely behaviour of the attackers and seeks to optimise the utility of the defender's decisions. In ARA, the decision problem is structured and represented by an Influence Diagram (ID) which generalizes Bayesian Networks. Here we show how a sequential defend-attack game model borrowed from Banks et al. (2015), can be accommodated to construct an EFBN.

An ID is a directed acyclic graph with three kinds of nodes: decision nodes, shown as rectangles; chance nodes, shown as ovals; and utility nodes, shown as diamonds. Fig. 13 shows the ID of a sequential defend-attack game model in the defender's perspective (Banks et al., 2015). In this model, the defender has a discrete set of possible defence levels $D = \{d_1, d_2, \ldots, d_n\}$, which are represented by the decision node D (Defences). After observing the potential defence levels that can be implemented, the attacker has a discrete set of possible attack levels $A = \{a_1, a_2, \ldots, a_m\}$ represented by node A (Attacks). A dashed arc pointing from node D to A represents the fact that the attacker's decision depends on the potential defences. Moreover, from the defender's perspective, the choice made by the attacker is a random variable. Hence, node A is a chance node rather than a decision node in this model. Whether the attack is successful is represented by the chance node S which is conditional on D and A. Finally, D and S determine D's utility.

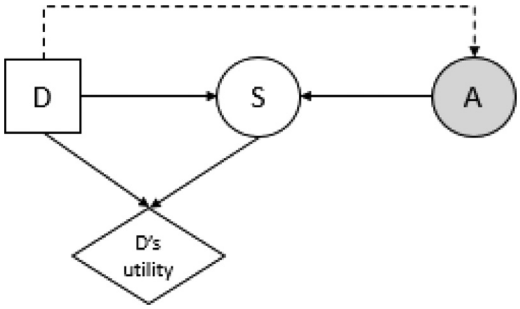
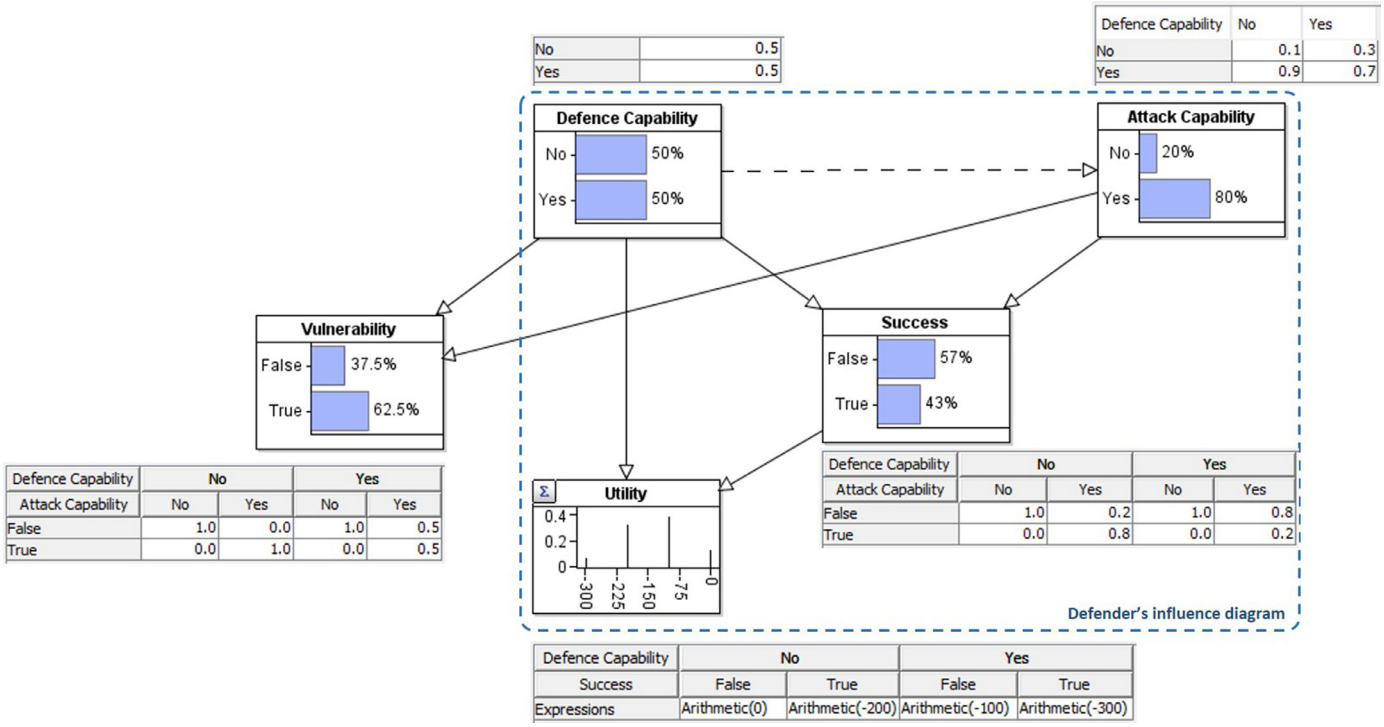


Fig. 13. The defender's influence diagram.

An example using adversarial risk analysis is shown in Fig. 14, which is represented by an Influence Diagram (ID) built with a BN. We assume that the defender's decision is about whether to equip the capability of a defence, d, to protect a target information asset. Meanwhile, after observing the defender's capability, the attacker would consider whether to deploy a corresponding attack



Defence Capability	No	Yes
No	0.5	
Yes	0.5	

Defence Capability	No		Yes	
Attack Capability	No	Yes	No	Yes
False	1.0	0.0	1.0	0.5
True	0.0	1.0	0.0	0.5

Defence Capability	No		Yes	
Attack Capability	No	Yes	No	Yes
False	1.0	0.2	1.0	0.8
True	0.0	0.8	0.0	0.2

Defence Capability	No		Yes	
Success	False	True	False	True
Expressions	Arithmetic(0)	Arithmetic(-200)	Arithmetic(-100)	Arithmetic(-300)

Defence Capability	No	Yes
No	0.1	0.3
Yes	0.9	0.7

Fig. 14. The BN according to the defender's ID.

capability, a, against it. Here we give uniform values to the Defence Capability node, representing the defender's open mindedness, while assuming that if the attacker finds that the defender has capability d, the probability that she deploys capability, a, is 0.9, otherwise, the probability under different circumstances would be lower (0.7). This is shown by the CPTs in Fig. 14. The CPT of the Success node models how the attacker and defence capabilities interact to determine the probability of attacker success. The utility node models the defender's payoff given the defence capability deployed (utility: −100) and the cost of being attacked successfully (utility: −200). Here we specify utilities using individual values as an example. The utilities can also be assigned by distributions in this ID.

Typically, the aim in decision analysis is to maximize the utility node of the supported decision maker. Corresponding to the ID shown in Fig. 14, a Decision Tree (DT) can be generated using AgenaRisk. We show this DT in Fig. 15(a). The applied algorithms and details for generating DTs from hybrid IDs in AgenaRisk are described in Fenton and Neil, 2018). The optimum decision is shown with the bold arc in the DT, showing the maximum utility decision for the defender is to deploy the defence capability (utility: −128, otherwise the utility would be −144). By entering this decision to the ID, we can assess vulnerability of the asset, shown in Fig. 15(b), which can be then used in our FAIR-BN for further analysis.

8. Discussion

We have introduced how we use BNs and the MC method to implement the calculation through the FAIR model and compared the performance of the three methods. In this section, we discuss performance, efficiency, flexibility, expandability features of the FAIR model, FAIR-BN and FAIR-MC from the perspective of cyber analysts and cyber risk managers.

First of all, in general, the three methods provide consistent results. However, the accuracy of the FAIR model is inevitably impaired by its tailored algorithms, and this inaccuracy becomes more obvious in certain cases, such as in long-tailed distribution scenarios. This is because the FAIR model uses triangular distributions to approximate input distributions and relies on cached data and interpolation for calculation. As we illustrated in Section 6.2, when LEF has the long-tail feature or LEF and LM follow other distributions, the FAIR accuracy decreases. In comparison, FAIR-BN can providestable and accurate results in general and in these specific cases. The calculation of FAIR-MC is intuitive and straightforward. To implement calculations through the FAIR model, which are listed in Table 1, FAIR-MC generates random samples following determined input distributions and operates these samples following the corresponding function to simulate the output variable. Since no other approximation techniques are applied, we assume a large number of samples generated by FAIR-MC can reflect the distribution of the output variable. Illustrated by the experimental results, FAIR-MC and FAIR-BN outperform the FAIR model in accuracy.

The three methods have identical efficiency in calculating LEF and its sub-factors. The FAIR model calculation is more efficient when performing risk aggregation compared with the other algorithms. This efficiency is achieved by the pre-processing of cached data which is calculated from $27 \times 12 \times 1,000,000$ samples that are generated by simulation (The Open Group, 2019). In comparison, FAIR-BN and FAIR-MC can still have comparable efficiency compared with the FAIR model in conducting the RA_1 process but require more calculation time in conducting the RA_2 process because each calculation is done anew for each case rather than reused from a cache. With more computational source available (i.e. Using GPU Clusters) and optimize the code efficiency, the gap of computational cost in the RA_2 process will further decrease.

The FAIR model and the proposed FAIR-BN do address "small data". The input of the FAIR model can be based on historical data, expertise or both of them, which makes "small data" acceptable. For example, in the FAIR model, the input of Primary Loss Event Frequency (PLEF) is a triangular distribution, whose parameters (lower bound, upper bound, and most likely value) can be as-

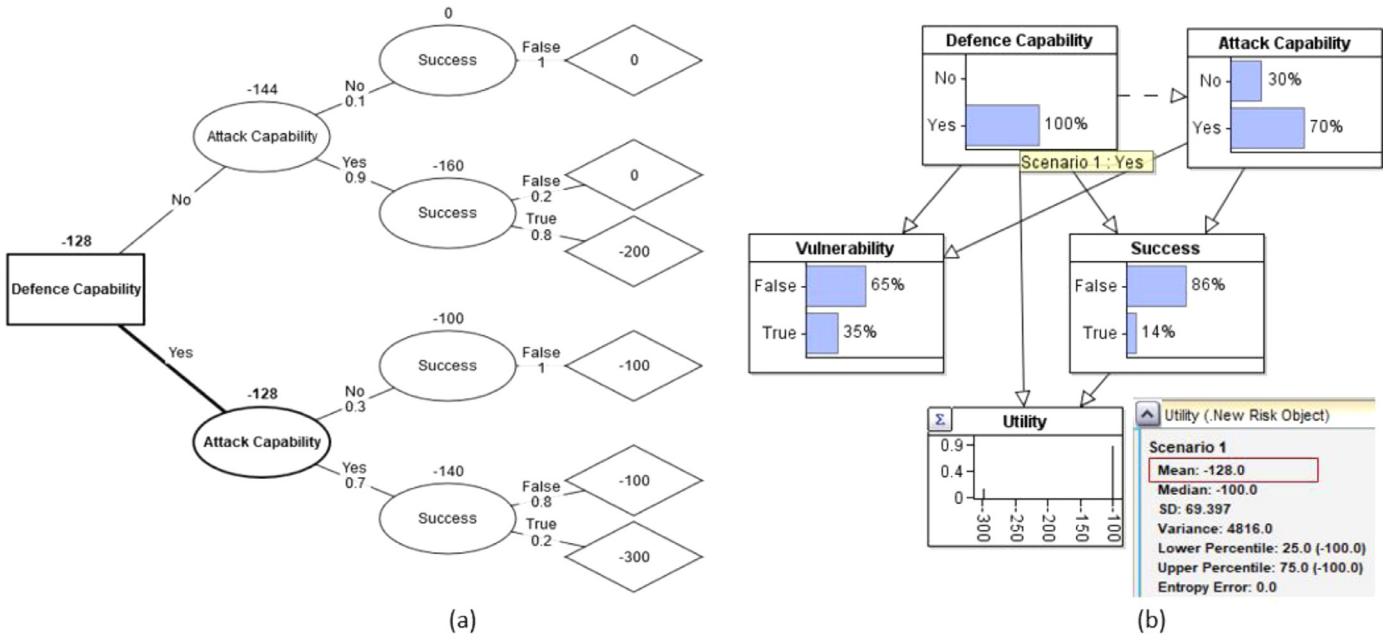


Fig. 15. Decision results of the defender's ID.

signed by historical data or by an expert's knowledge. Large data is not a necessary condition here: if a loss event happens five times a year, parameters of the triangular distribution can still be determined based on this frequency and adjusted by an expert. The FAIR-BN can similarly specify inputs using small data. Moreover, in the FAIR-BN, there is more flexibility, since there is no limitation on the input distributions.

In practice, risk factors (i.e. LEF and LM) can have diverse features, but the algorithms of FAIR are based on the precondition that input variables follow triangular distributions. Otherwise, cached data and the application of the BMD function (see Appendix B) become invalid. In contrast, the FAIR-BN and FAIR-MC employ more flexible algorithms which do not have limitations of input. Calculations for both FAIR and FAIR-MC are based on sampling, which provides no modularized modelling mechanism; hence neither FAIR nor FAIR-MC are easily extendable with other mature CRA models for risk assessment and decision making. In comparison, FAIR-BN can easily incorporate other dedicated CRA models, which is significant in practice. We have illustrated the expandability of FAIR-BN by extending it using a process-oriented model and a defend-attack game model in Section 7.

The three methods all have their pros and cons. When preliminary and high-level risk assessment is required, where efficiency is prioritized over the accuracy, the FAIR model would be the preferable choice. FAIR-MC is more suitable in cases where greater accuracy is required, but no further modular extension of the model is needed. FAIR-BN would be the best choice if risk managers or researchers require higher result accuracy, modular expandability of the model for more detailed analysis, and integrated decision supporting.

9. Conclusion

The FAIR model provides both a methodology and a tool for cybersecurity risk analysis and calculation. It is an ideal choice for conducting risk assessment where the focus is on calculating expected economic loss arising from cybersecurity risk. However, FAIR makes inflexible assumptions that limit both its accuracy for a range of real-world scenarios and the possibility of integrating it into other mature CRA models. We have revealed the structure underlying FAIR and tested it against algorithmic alternatives in the form of (a) an MC version of FAIR (FAIR-MC) and (b) a BN version (FAIR-BN). Experimental results show that, when we adopt the FAIR model's underlying assumptions and input distribution requirements, both FAIR and FAIR-BN produce favourable results when compared with FAIR-MC. However, the FAIR model provides less accurate results in a number of scenarios, primarily where we have a long-tailed distribution. Hence, the approximation approach embedded within FAIR improves efficiency but at a cost in accuracy. In comparison, FAIR-BN provides more stable performance in result accuracy across a wider set of scenarios involving widely varied distributions, but at a cost in efficiency.

As well as carrying out an empirical evaluation of FAIR we have also analysed the rigidity of FAIR and shown how it can be extended, using FAIR-BN as the foundation, to cope with more diverse distributions and statistical functions, but also, more importantly, to accommodate causal reasoning for modelling richer defend-attack contexts. We have illustrated this by constructing the Extended FAIR-BNs (EFBNs) incorporating a process-oriented model and a defend-attack game model. EFBN can model relevant knowledge about the causal processes that give risk to cyber events and the likely economic consequences of such events and do so in a way that is consistent and compatible with the FAIR model. Based on these results, our future research will focus on promoting advanced EFBN, from both process-oriented and game-theoretic perspectives, and exploring constructing EFBN from data.

Acknowledgments

Jiali Wang is supported by a China Scholarship Council (CSC)/Queen Mary Joint Ph.D. scholarship. Martin Neil and Norman Fenton were partly supported in this work by the Leverhulme Trust under project CAUSAL-DYNAMICS. We are grateful to the editor as well as the anonymous reviewers for comments and suggestions on the paper. Agena Ltd provided the AgenaRisk software gratis.

Appendix A: The Bounded Metalog Distribution

A Bounded Metalog Distribution (BMD) is a quantile function of a random variable M. A BMD can be specified by distinct quantile

points on the Cumulative Density Function (CDF) of M, and then is used to simulate samples of M stochastically in the FAIR model, by inputting randomly generated probabilities (from 0 to 1) into its expression. Constructing the BMD of the total loss variable is the core of how risk aggregation is effectively conducted in the FAIR model. Since BMD is derived from its general version, Metalog Distribution (MD) (Keelin, 2016), which does not have lower or upper bound, we start from the MD to explain the BMD.

Given n distinct quantile points on the CDF of a random variable, the corresponding n-term MD can be uniquely specified. The formal definition is described as below.

Definition 1 (Keelin, 2016): The Metalog distribution of a random variable M with n terms is shown by formula (A.1):

$$M_n(y; x,\ y) = \begin{cases} a_1 + a_2 ln\left(\frac{y}{1-y}\right) & \text{for } n = 2 \\ a_1 + a_2 ln\left(\frac{y}{1-y}\right) + a_3(y-0.5) ln\left(\frac{y}{1-y}\right) & \text{for } n = 3 \\ a_1 + a_2 ln\left(\frac{y}{1-y}\right) + a_3(y-0.5) ln\left(\frac{y}{1-y}\right) + a_4(y-0.5) & \text{for } n = 4 \\ M_{n-1} + a_n(y-0.5)^{\frac{n-1}{2}} & \text{for odd } n \geq 5 \\ M_{n-1} + a_n(y-0.5)^{\frac{n}{2}-1} ln\left(\frac{y}{1-y}\right) & \text{for even } n \geq 6 \end{cases} \quad (A.1)$$

Where y is a cumulative probability with $0 < y < 1$. Column vectors $\mathbf{x} = (x_1, \ldots, x_m)$ and $\mathbf{y} = (y_1, \ldots, y_m)$ are of length $m(\geq n)$. Each pair of $(x_i,\ y_i)$ represents a point on the CDF of the random variable M, with $0 < y_i < 1$, and at least n of y_i are distinct. The column vector of scaling constants $\mathbf{a} = (a_1, \ldots, a_n)$ is given by formula (A.2)

$$\mathbf{a} = \left[\mathbf{Y}_n^T \mathbf{Y}_n\right]^{-1} \mathbf{Y}_n^T \mathbf{x} \quad (A.2)$$

where $\mathbf{Y}_n^T$ is the transpose of $\mathbf{Y}_n$, whilst the $m \times n$ matrix $\mathbf{Y}_n$ is shown by (A.3):

$$\mathbf{Y}_n = \begin{cases} \begin{bmatrix} 1 & ln\left(\frac{y_1}{1-y_1}\right) \\ \vdots & \\ 1 & ln\left(\frac{y_m}{1-y_m}\right) \end{bmatrix} & \text{for } n = 2 \\ \begin{bmatrix} 1 & ln\left(\frac{y_1}{1-y_1}\right) & (y_1-0.5) ln\left(\frac{y_1}{1-y_1}\right) \\ & \vdots & \\ 1 & ln\left(\frac{y_m}{1-y_m}\right) & (y_m-0.5) ln\left(\frac{y_m}{1-y_m}\right) \end{bmatrix} & \text{for } n = 3 \\ \begin{bmatrix} 1 & ln\left(\frac{y_1}{1-y_1}\right) & (y_1-0.5) ln\left(\frac{y_1}{1-y_1}\right) & (y_1-0.5) \\ & \vdots & & \\ 1 & ln\left(\frac{y_m}{1-y_m}\right) & (y_m-0.5) ln\left(\frac{y_m}{1-y_m}\right) & (y_m-0.5) \end{bmatrix} & \text{for } n = 4 \\ \begin{bmatrix} & (y_1-0.5)^{\frac{n-1}{2}} \\ \mathbf{Y}_{n-1} | & \vdots \\ & (y_m-0.5)^{\frac{n-1}{2}} \end{bmatrix} & \text{for odd } n \geq 5 \\ \begin{bmatrix} & (y_1-0.5)^{\frac{n}{2}-1} ln\left(\frac{y_1}{1-y_1}\right) \\ \mathbf{Y}_{n-1} | & \vdots \\ & (y_m-0.5)^{\frac{n}{2}-1} ln\left(\frac{y_m}{1-y_m}\right) \end{bmatrix} & \text{for even } n \geq 6 \end{cases} \quad (A.3)$$

The proof that the quantile function of a random variable M can be parameterized by points on the CDF of M is provided in Keelin (2016).

The Bounded Metalog Distribution is defined based on Metalog Distribution as below:

Definition 2 (Keelin, 2016): Bounded Metalog Distribution (BMD)

A BMD is a modified Melalog distribution which has known lower and upper bounds, b_l and b_u respectively, with $b_l < b_u$. It is also called the logit Metalog distribution. The BMD is the transformation of a Metalog distribution, in which $z = ln(\frac{x-b_l}{b_u-x})$ is Metalog-distributed.

Setting $ln(\frac{x-b_l}{b_u-x})$ equal to (A.1) and solving for x yields, the BMD function with n terms can be obtained from (A.4):

$$M_n^{logit}(y; x,\ y,\ b_l, b_u) = \begin{cases} \frac{b_l + b_u e^{M_n(y)}}{1 + e^{M_n(y)}} & 0 < y < 1 \\ b_l & y = 0 \\ b_u & y = 1 \end{cases} \quad (A.4)$$

In the FAIR model, the quantile function of the total loss variable is represented by the BMD, which is constructed using cached quantile values. Then by randomly generating a probability, y, and substituting it in formula (A.4), a sample of the total loss can be simulated. This is the basic idea of how BMD is implemented to efficiently simulate losses. We explain it formally and technically in Appendix B.

Appendix B: Application of BMD in Risk Aggregation

In the FAIR model, Primary Losses (PL) and Secondary Losses (SL) are simulated using the same risk aggregation method. Here we use PL as the example to explain how risk aggregation is implemented in the FAIR model. Firstly, a large amount of PL samples, L_P, are simulated associating with predetermined Frequencies (F), $\hat{f}_j$, and different shape modes of Loss Magnitudes (LM), $\hat{s}_k$, using an MC method in advance. Here $\hat{f}_j \in F$, with j from 0 to 27, and F is a set of a few predetermined frequencies covering 0 to 1001 (The FAIR model assumes that when the frequency is larger than 1001, distributions of L_P would converge to normal distributions. Therefore, L_P can be represented by normal distributions directly rather than using risk aggregation to generate its samples). Moreover, the FAIR model introduces a concept, shape mode, classifying all the triangular distributions into 12 shape modes. The shape mode, $\hat{s}_k$, represents the ratio $r = \frac{M_{ml} - M_{\min}}{M_{\max} - M_{\min}}$, and is classified into a set of predetermined ratios, $S = \{0, 0.1, 0.2, \ldots, 0.9, 1, 1.01\}$. These L_P samples are firstly taken to average over the corresponding $\hat{f}_i$, and then used to generate Cumulative Density Functions (CDF) of average samples, $\bar{L}_P$, corresponding to each pair of $(\hat{f}_j,\ \hat{s}_k)$. The quantile value vector, $\mathbf{v} = (v_1, \ldots, v_9)$, associated with nine predetermined quantile probabilities, $\mathbf{y} = (0.001,\ 0.01,\ 0.1,\ 0.25,\ 0.5,\ 0.75,\ 0.9,\ 0.99,\ 0.999)$, on each CDF can then be calculated and are cached as a vector. By doing so, a 27 ×12 sized data matrix is produced. Each element of this matrix is a vector, v, corresponding to a pair of $(\hat{f}_j,\ \hat{s}_k)$. This data matrix is prepared and provided by the FAIR model (The Open Group, 2019).

Based on the cached data, the FAIR model approximates the quantile value vector, v, for the actual frequency sample f_i and a LM distribution of ratio, r, by applying interpolation on cached vectors, of which the corresponding $\hat{f}_j$ and $\hat{s}_k$ are close to f_i and r. We extract the interpolation formula from the FAIR model and show it by formula (B.1):

$$v = \left(In\left(\frac{v_1}{1-v_1}\right) \times a + In\left(\frac{v_2}{1-v_2}\right) \times (1-a)\right) \times b + \left(In\left(\frac{v_3}{1-v_3}\right) \times a + In\left(\frac{v_3}{1-v_3}\right) \times (1-a)\right) \times (1-b) \quad (B.1)$$

Where $a = \frac{f_{max} - f_i}{f_{max} - f_{min}}$ and $b = \frac{r_{max} - r}{r_{max} - r_{min}}$

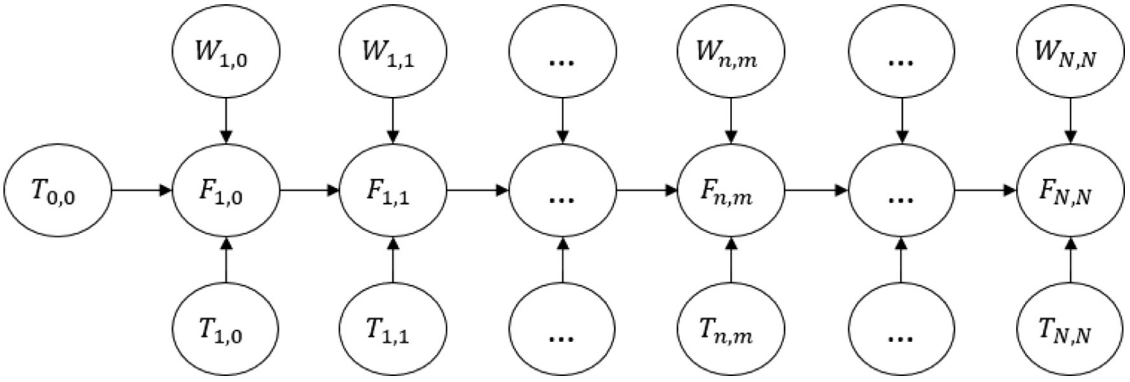


Fig. 16. Factorization of the BN in the RA_2 process.

In formula (B.1), v is the quantile value vector which stores approximated quantile values corresponding to f_i and r, while v_1, v_2 and v_3 are corresponding to (f_{min}, r_{min}), (f_{max}, r_{min}) and $(f_{min}, r_{\max})$ respectively. The frequency, f_{max}, is the frequency in the predetermined frequency set, F, which is close to and larger than f_i, while f_{min} is the frequency in F, which is close to and smaller than f_i. The ratio r, which calculated by $\frac{M_{ml}-M_{\min}}{M_{\max}-M_{\min}}$, represents the actual shape mode of a triangular distribution; r_{max} is the shape ratio in S, which is close to and larger than r, while r_{min} is the shape ratio in S, which is close to and smaller than r.

Therefore, for each pair of (f_i, r), the quantile value vector of the corresponding $\bar{L}_P$ can be approximated using cached data $(v_1, v_2$ and $v_3)$ following formula (B.1). The approximated v is then used to specify the Metalog distribution (Keelin, 2016) of $\bar{L}_P$. The Metalog distribution is a kind of logistic quantile distribution that can be determined by quantile values. For example, v, which contains nine quantile values, can be used to specify a 9-term Metalog distribution of $\bar{L}_P$. We denote this distribution as $M_9(y)$. Assigning a uniformly generated probability to y, a logistic sample of $\bar{L}_P$ can be calculated by $M_9(y)$. Since $M_9(y)$ represents the logistic sample of $\bar{L}_P$ related to (f_i, r), the sample of L_P, $L_P(i)$, can be generated by taking exponent and changing scale of $M_9(y)$ following formula (B.2), which is referred to as Bounded Metalog Distribution (BMD) in Keelin (2016). We have described details of Metalog distribution and BMD in Appendix A.

$$L_P(i) = f_i * \left(M_{min} + M_{max} \frac{e^{M_9(y)}}{1 + e^{M_9(y)}} \right) \tag{B.2}$$

In conclusion, the core mechanism of conducting risk aggregation in the FAIR model is to construct BMDs of the given (F_P, LM_P). More precisely, for each sample of F_P, f_i, a BMD is specified using cached data and is then used to generate a sample of primary loss, $L_P(i)$, by substituting y using a uniformly generated probability in formula (B.2). By this way, the sample vector of L_P is generated. By now, we have explained how risk aggregation, RA, is implemented to simulate primary losses in the FAIR model. We denote this simulation by $L_P = RA(F_P, LM_P)$. In addition, the FAIR model does not distinguish risk aggregation of simulating primary losses and secondary losses. In other words, secondary losses are simulated following the same way which can be represented by $L_S = RA(F_S, LM_S)$, where F_S and LM_S represent frequencies and loss magnitudes of secondary losses respectively. Furthermore, the Total Loss, L_T, is simulated by $L_T = L_P + L_S$.

Appendix C: Factorization of the BN for RA_2 process

We demonstrate the adjusted Compound Density Factorization (CDF) method in Fig. 16. Since each total loss variable $T_{n,m}$ is mutually exclusive, i.e. for a given value of N, the sum of probabilities related to each possible scenario is equal to one, we factorize the BN (e) by introducing extra two kinds of variables. Boolean variables, $W_{n,m}$ (with only two states True and False) are used to assign weightings proportional to $P_{n,m}$, to each pair of nodes, i.e. $\{T_{0,0}, T_{1,0}\}$, $\{F_{1,0}, T_{1,1}\}$, ..., $\{F_{N,N-1}, T_{N,N}\}$. Factor variables, $F_{n,m}$, are created to calculate the weighted aggregation for each step.

The Conditional Probability Table (CPT) for $W_{n,m}$ is defined by formula (C.1):

$$P(W_{n,m-1} = true) = \frac{P_{0,0} + P_{1,0} + \ldots + P_{n,m-1}}{P_{0,0} + P_{1,0} + \ldots + P_{n,m}} \tag{C.1}$$

The conditionally deterministic expression for variable $F_{n,m}$, which is called a partitioned node in the BN parlance, is defined by formula (C.2):

$$F_{n,m} = \begin{cases} F_{n,m-1} & \text{if } W_{n,m} = True \\ T_{n,m} & \text{if } W_{n,m} = False \end{cases} \tag{C.2}$$

Since $T_{0,0}$ and $T_{1,0}$ are mutually exclusive, the marginal distribution for variable $F_{1,0}$ is represented by formula (C.3):

$$F_{1,0} = P(W_{1,0} = True)\ T_{0,0} + P(W_{1,0} = False)\ T_{1,0} \tag{C.3}$$

Similarly, the marginal for variable $F_{n,m}$ is represented by formula (C.4):

$$F_{n,m} = P(W_{n,m} = True)\ F_{n,m-1} + P(W_{n,m} = False)\ T_{n,m} \tag{C.4}$$

After factorizing the density aggregation process, we can calculate the marginal distribution of $F_{n,m}$ more efficiently following formula (C.4), which yields the risk aggregation result given primary and secondary loss frequencies and their loss magnitudes. We have implemented this method using AgenaRisk packages.

Appendix D: Experimental Results for Subsidiary Risk Factors

Table 8
Comparison results of $M_{PLEF} = F_{TE} \times P_V$ with P_V following *Triangular*(min = 0.2, ml = 0.3, max = 0.7) (the calculation is similar with $F_{TE} = F_C \times P_A$).

Test	Description	FTE			Mean			Variance			99th			J(FAIR, FAIR-MC)	J(FAIR-BN, FAIR-MC)
		min	mid	max	FAIR	FAIR-BN	FAIR-MC	FAIR	FAIR-BN	FAIR-MC	FAIR	FAIR-BN	FAIR-MC		
1	Include 0	0	200	500	92.5	93.8	93.3	2329.5	2519.6	2442.9	226.1	241.8	236.3	0.0348	0.0057
2	Long tail	100	200	1000	171.7	173.0	173.2	8570.6	9095.6	9096.7	440.3	456.4	462.0	0.0272	0.0063
3	Left skew	20	80	200	39.7	40.0	40.0	338.2	364.0	356.6	91.7	96.6	95.6	0.0217	0.0058
4	Right skew	20	160	200	50.5	50.7	50.7	431.3	448.3	443.2	102.7	105.4	105.8	0.0453	0.0090
5	0 and long tail	0	200	1000	158.1	159.9	159.9	9287.4	9753.9	9866.9	433.1	458.2	458.4	0.0260	0.0077
													Average:	0.0310	0.0069

Table 9
Comparison results of F_P = Poisson($\lambda = M_{PLEF}$).

Test	Description	MPLEF			Mean			Variance			99th			J(FAIR, FAIR-MC)	J(FAIR-BN,FAIR-MC)
		min	mid	max	FAIR	FAIR-BN	FAIR-MC	FAIR	FAIR-BN	FAIR-MC	FAIR	FAIR-BN	FAIR-MC		
1	Include 0	0	200	500	237.7	232.6	233.3	10219.0	10780.0	10788.0	466.0	464.6	468.0	0.0106	0.0024
2	Long tail	100	200	1000	441.1	434.5	433.2	39325.0	41563.0	41027.0	921.0	464.6	920.0	0.0162	0.0027
3	Left skew	20	80	200	101.2	99.8	100.0	1437.6	1487.6	1501.9	191.5	464.6	192.0	0.0163	0.0067
4	Right skew	20	160	200	128.5	126.5	126.6	1520.3	1624.4	1616.4	197.5	202.4	202.0	0.0272	0.0130
5	0 and long tail	0	200	1000	408.6	400.3	400.0	44983.0	47421.0	47035.0	916.0	908.5	915.0	0.0148	0.0046
													Average:	0.0170	0.0059

Table 10
Comparison results of $F_S = Binomial(n = F_P, P = P_{SL})$ with P_{SL} following *Triangular*(min = 0.25, ml = 0.3, max = 0.7).

Test	Description	MPLEF			Mean			Variance			99th			J(FAIR, FAIR-MC)	J(FAIR-BN, FAIR-MC)
		min	mid	max	FAIR	FAIR-BN	FAIR-MC	FAIR	FAIR-BN	FAIR-MC	FAIR	FAIR-BN	FAIR-MC		
1	Include 0	0	200	500	95.3	93.6	93.4	2401.3	2562.1	2543.0	231.5	238.2	240.0	0.0127	0.0041
2	Long tail	100	200	1000	176.7	173.6	173.4	8796.3	9451.9	9330.9	452.0	469.1	466.0	0.0231	0.0029
3	Left skew	20	80	200	40.6	40.2	40.0	381.9	405.3	397.5	97.5	99.5	99.0	0.0127	0.0058
4	Right skew	20	160	200	51.7	50.7	50.7	479.3	497.2	493.7	109.0	108.4	110.0	0.0355	0.0066
5	0 and long tail	0	200	1000	163.6	160.5	159.9	9445.9	10249.0	10017.0	447.0	463.4	461.0	0.0223	0.0069
													Average:	0.0213	0.0053

Table 11
Results of $P_V = P(P_{TC} > P_{RS})$ with P_{RS} following*Triangular*(min = 0.2, ml = 0.3, max = 0.7).

Test	description	PTC			PV		
		min	mid	max	FAIR	FAIR-BN	FAIR-MC
1	Include 0	0.00	0.20	0.50	0.17	0.14	0.14
2	Long tail	0.10	0.20	1.00	0.51	0.52	0.52
3	l eft s kew	0.20	0.40	0.80	0.62	0.65	0.65
4	right skew	0.20	0.60	0.80	0.75	0.78	0.79
5	0 and long tail	0.00	0.20	1.00	0.46	0.47	0.46

Appendix E: Experimental Results with Euclidean Distance Measurement

Table 12
Results comparison of L_P distributions with inputs following triangular distributions - the Euclidean distance measurement.

Test	MPLEF			Eu(FAIR, FAIR-MC)	Eu(FAIR-BN, FAIR-MC)
	min	mid	max		
1	0	20	90	0.0232	0.0202
2	0	230	300	0.0387	0.0224
3	20	80	180	0.0306	0.0213
4	60	250	400	0.0339	0.0355
5	20	250	630	0.0241	0.0211
6	15	30	250	0.0216	0.0203
7	15	30	540	0.0263	0.0218
			Average:	0.0283	0.0232

Table 13
Results comparison of L_T distributions - the Euclidean distance measurement.

Test	**MPLEF**			Eu(FAIR, FAIR-MC)	Eu(FAIR-BN, FAIR-MC)
	min	mid	max		
1	0	200	500	0.0361	0.0128
2	50	200	1000	0.0448	0.0160
3	20	80	200	0.0331	0.0304
4	20	160	200	0.0415	0.0149
5	0	200	1000	0.0337	0.0151
			Average:	0.0378	0.0178

Table 14
Results comparison of L_P distributions with F_P following long-tailed distributions - the Euclidean distance measurement.

Test	Input Distributions		Eu(FAIR, FAIR-MC)	Eu(FAIR-BN, FAIR-MC)
	PLEF	PLM		
1	Weibull	LogNormal	0.1822	0.0166
2	Log Normal	LogNormal	0.2745	0.0147
3	Gamma	LogNormal	0.1799	0.0177
		Average:	0.2122	0.0163

References

Cashell, B., et al., 2004. *The Economic Impact of cyber-attacks*. Congressional Research Service Documents. CRS, Washington DC RL32331.

Hutchins, E.M., Cloppert, M.J., Amin, R.M., 2011. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. Leading Issues in Information Warfare & Security Research, 1. Academic Publishing International, p. 80.

Blakley, B., McDermott, E., Geer, D., 2001. Information security is information risk management. In: Proceedings of the Workshop on New Security Paradigms. ACM.

Peltier, T.R., 2010. Information Security Risk Analysis. Auerbach publications.

Freund, J., Jones, J., 2014. Measuring and Managing Information Risk: A FAIR Approach. Butterworth-Heinemann.

Jones, J., 2006. An introduction to factor analysis of information risk (fair). Norwich J. Info. Assur. 2 (1), 67.

Sendi, A.S., Cheriet, M., 2014. Cloud computing: a risk assessment model. In: Proceedings of the IEEE International Conference on Cloud Engineering. IEEE.

Le, A., et al., 2017. Assessing loss event frequencies of smart grid cyber threats: encoding flexibility into fair using bayesian network approach. In: Smart Grid Inspired Future Technologies. Springer, pp. 43–51.

Park, M., et al., 2018. Situational awareness framework for threat intelligence measurement of android malware. JoWUA 9 (3), 25–38.

Wangen, G., Hallstensen, C., Snekkenes, E., 2016. A framework for estimating information security risk assessment method completeness. Int. J. Inf. Secur. 1–19.

Fenton, N., Neil, M., 2018. Risk Assessment and Decision Analysis with Bayesian Networks. Crc Press.

Foss, S., Korshunov, D., Zachary, S., 2013. Heavy-tailed and long-tailed distributions. An Introduction to Heavy-Tailed and Subexponential Distributions. Springer, pp. 7–42.

Nielsen, T.D., Jensen, F.V., 2009. Bayesian Networks and Decision Graphs. Springer Science & Business Media.

Poolsappasit, N., Dewri, R., Ray, I., 2012. Dynamic security risk management using Bayesian attack graphs. IEEE Trans. Depend. Secure. Comput. 9 (1), 61–74.

Banks, D.L., Aliaga, J.M.R., Insua, D.R., 2015. Adversarial Risk Analysis. Chapman and Hall/CRC.

IBM, 2018. Cost of Data Breach Study: Impact of Business Continuity Management. Ponemon Institute LLC.

Jeffreys, H., 1946. An invariant form for the prior probability in estimation problems. In: Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences, pp. 453–461.

Lin, J., 1991. Divergence measures based on the Shannon entropy. IEEE Trans. Inf. Theory 37 (1), 145–151.

Keelin, T.W., 2016. The metalog distributions. Decis. Anal. 13 (4), 243–277.
Le, A., et al., 2019. Incorporating FAIR into Bayesian Network for numerical assessment of loss event frequencies of smart grid cyber threats Mob. Netw. Appl. 24, 1713–1721.
LeBlanc, D., Howard, M., 2002. Writing Secure Code. Pearson Education.
Schneier, B., 1999. Attack trees. Dr. Dobb's J. 24 (12), 21–29.
Jha, S., Sheyner, O., Wing, J., 2002. Two formal analyses of attack graphs. In: Proceedings of the 15th IEEE Computer Security Foundations Workshop. IEEE.
Bistarelli, S., Fioravanti, F., Peretti, P., 2006. Defense trees for economic evaluation of security investments. In: Proceedings of the First International Conference on Availability, Reliability and Security. ARES 2006. IEEE.
Liu, Y., Man, H., 2005. Network vulnerability assessment using Bayesian networks. Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security. International Society for Optics and Photonics.
Xie, P., et al., 2010. Using bayesian networks for cyber security analysis. In: Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). IEEE.
Lin, P., Neil, M., Fenton, N., 2014. Risk aggregation in the presence of discrete causally connected random variables. Ann. Actuar. Sci. 8 (2), 298–319.
Manshaei, M.H., et al., 2013. Game theory meets network security and privacy. ACM Comput. Surv. (CSUR) 45 (3), 25.
Do, C.T., et al., 2017. Game theory for cyber security and privacy. ACM Comput. Surv. (CSUR) 50 (2), 30.
Roy, S., et al., 2010. A survey of game theory as applied to network security. In: Proceedings of the 43rd Hawaii International Conference on System Sciences. IEEE.
Wang, Y., et al., 2016. A survey of game theoretic methods for cyber security. In: Proceedings of the IEEE First International Conference on Data Science in Cyberspace (DSC). IEEE.
Lye, K.-w, Wing, J.M., 2005. Game strategies in network security. Int. J. Inf. Secur. 4 (1–2), 71–86.
Nguyen, K.C., Alpcan, T., Basar, T., 2009. Security games with incomplete information. In: Proceedings of the IEEE International Conference on Communications. IEEE.
Rass, S., König, S., Schauer, S., 2015. Uncertainty in games: using probability-distributions as payoffs. In: Proceedings of the International Conference on Decision and Game Theory for Security. Springer.
Rass, S., König, S., Schauer, S., 2017. Defending against advanced persistent threats using game-theory. PLoS ONE 12 (1), e0168675.
Brown, G.G., Carlyle, W.M., Wood, R.K., 2008. Optimizing Department of Homeland Security Defense investments: Applying defender-Attacker (-defender) Optimization to Terror Risk Assessment and Mitigation. Naval Postgraduate School Monterey CA Dept Of Operations Research.
Alderson, D.L., et al., 2011. Solving Defender-Attacker-Defender Models For Infrastructure Defense. Naval Postgraduate School Monterey CA Dept Of Operations Research.
Koller, D., Milch, B., 2003. Multi-agent influence diagrams for representing and solving games. Games Econ. Behav. 45 (1), 181–221.
Gruber, A., Ben-Gal, I., 2019. A targeted Bayesian network learning for classification. Qual. Technol. Quant. Manag. 16 (3), 243–261.
Zhu, M., Liu, S., Jiang, J., 2016. A hybrid method for learning multi-dimensional Bayesian network classifiers based on an optimization model. Appl. Intell. 44 (1), 123–148.
Heckman, P.E., Meyers, G.G., 1983. The calculation of aggregate loss distributions from claim severity and claim count distributions. In: Casualty Actuarial Society (Ed.), Proceedings of the Casualty Actuarial Society, 70. Casualty Actuarial Society, pp. 49–66.
MATLAB, 2018. *version 9.5.0 (R2018b)*, Natick, Massachusetts: The MathWorks Inc.
Anderson, C. 2004. The long tail. Wired Magazine (October) 170–177.
Zhou, Y., Fenton, N., Neil, M., 2014. Bayesian network approach to multinomial parameter learning using data and expert judgments. Int. J. Approx. Reason. 55 (5), 1252–1268.
Zhou, Y., et al., 2013. Incorporating expert judgement into Bayesian network machine learning. In: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence.
Fenton, N., Neil, M., 2011. The use of bayes and causal modelling in decision making, uncertainty and risk. CEPIS Upgrade 12 (5), 10–21.
The Open Group. 2019. The Open FAIR™ RISK ANALYSIS TOOL, https://publications.opengroup.org/i181. 2019.
The Build Security In initiative of the United States department of homeland security: https://buildsecurityin.us-cert.gov/bsi/articles/best-practices/deployment/583-BSI.html. 2019.
Agena Ltd.*2002-2019. AgenaRiskV10 software package*, www.AgenaRisk.com. 2019.
Mahadevan, S., 1997. Monte Carlo Simulation. Mechanical Engineering-New York and Basel-Marcel Dekker (1997), 123–146.
Kullback, S., Leibler, R.A., 1951. On information and sufficiency. Ann. Math. Stat. 22 (1), 79–86.
Tankard, C., 2011. Advanced persistent threats and how to monitor and deter them. Netw. Secur. 2011 (8), 16–19.
Danielsson, P.-E., 1980. Euclidean distance mapping. Comput. Gr. Image Process. 14 (3), 227–248.
Robert, C.P., 1995. Simulation of truncated normal variables. Stat. Comput. 5 (2), 121–125.
Rios Insua, D., Rios, J., Banks, D., 2009. Adversarial risk analysis. J. Am. Stat. Assoc. 104 (486), 841–854.
The Open Group. 2018. The Open FAIR™ tool with Sipmath™ Distributions guide to the theory of operation, https://publications.opengroup.org/g181. 2019.
ISACA, The Risk IT Framework. http://www.isaca.org/Knowledge-Center/Research/ResearchDeliverables/Pages/The-Risk-IT-Framework.aspx. 2019.

Jiali Wang: Jiali is a Ph.D. student in Computer Science at Queen Mary University. She got her MSc in Quantitative Finance at Lancaster University, UK, and B.S in Mathematics and Applied Mathematics at the University of Science and Technology Beijing, China. Her research has focused on Bayesian modelling, intelligent risk assessment, decision support, machine learning, and cybersecurity risk assessment.

Martin Neil: Martin is Professor in Computer Science and Statistics in Queen Mary University of London and a Director of Agena Ltd. His research interests cover Bayesian modelling, intelligent risk assessment and decision analysis in diverse areas including systems and design reliability, project risk, decision support, cost-benefit analysis, AI and personalization, machine learning, and cyber security. Martin is also a fellow of the Alan Turing Institute.

Norman Fenton: Norman Fenton is a Director of Agena and also Professor of Risk Information Management at Queen Mary London University. Norman is a mathematician by training whose current research focuses on critical decision-making and, in particular, on quantifying uncertainty. Norman has published 7 books (covering risk, probability, maths and software) and 280 referred papers. His work includes applications in law and forensics (he has been an expert witness in many major criminal and civil cases), medicine, security, software reliability, transport safety and reliability, finance, and football prediction.