A Bayesian-Network-Based Framework for Risk Analysis and Decision Making in Cybersecurity

Jiali Wang

School of Electronic Engineering & Computer Science

Queen Mary University of London.

This dissertation is submitted for the degree of

Doctor of Philosophy

Submitted: July 2021

Amended: September 2021

Declaration

I, Jiali Wang, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged, and my contribution indicated. Previously published material is also acknowledged.

I attest that I have exercised reasonable care to ensure that the work is original and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

Jiali Wang

Submitted: July 2021

To my parents

Acknowledgement

Firstly, I would like to express my greatest gratitude to my principal supervisor, Professor Martin Neil. He provided me with enlightened guidance, continuous encouragement and support, and sufficient space for independent thinking during my whole Ph.D. study period. His rigorous but open attitude towards academic study has left a deep impact on me. I believe this positive influence will accompany me throughout my future work and life.

Many thanks to other panel members of my research: Professor Norman Fenton and Dr William Marsh for their insightful comments and suggestions for my research.

My warm appreciation goes to all the current and former researchers of the RIM group and especially Dr Haoyuan Zhang, Dr Peng Lin, Dr Christopher Joyner, Dr Scott McLachlan, Dr Eugene Dementiev, Dr Maggie Wang, Dr Evangelia Kyrimi, Mariana Raniere and Ali Fahmi for their kindness and help. I would like to thank EECS administrative staff, especially Mellisa Yao, for taking care of my Ph.D. progress.

I would like to express my deepest and sincere gratitude to my beloved parents, Jian and Yali for their unconditional love. No matter in my study, work or when I decided to pursue a Ph.D., they always gave me firm trust and support. Last but not the least, I would like to give special thanks to my beloved husband, Dr Xu Lan, for his constant love and support, especially during our Ph.D. study period together at QMUL.

Abstract

Many approaches have been proposed to define, measure and manage cybersecurity risk. A common theme underpinning Cybersecurity Risk Assessment (CRA) involves modelling relationships between risk factors and the use of statistical and probabilistic inference to calculate risk. This thesis focuses on the use of Bayesian Networks (BNs) for this dual purpose. The application of BNs to CRA was a non-trivial task while with the computational efficiency and flexibility of BN algorithms has improved such that they can now be widely applied to solve a variety of CRA problems. One such advance is in Hybrid Bayesian Networks (HBNs) to support inference in models containing discrete and continuous variables. HBNs are now routinely used for prediction and diagnostic inference tasks and have been extended, in the form of Influence Diagrams (IDs), to support decision making tasks.

This thesis proposes an HBN based CRA framework for comprehensive cybersecurity causal risk analysis and probabilistic calculation. We introduce causal risk analysis into cybersecurity problems and use a kill chain model to illustrate how causal analysis can guide the cybersecurity risk modelling. The proposed framework is flexible and extensible in a way that it can incorporate other CRA models built using BNs. We illustrate this by showing how the framework can incorporate risk analysis models of both organizational and technical perspectives. For organizational risk analysis, where the focus is on defending information assets/systems of organizations in an economically efficient way, the thesis shows how BNs can be used for modelling causal/probabilistic relationship between involved variables and conducting risk assessment. For technical risk analysis, which is motived by the perspective of cybersecurity analysts, it argues that IDs can be used to model the game between the defender and the attacker in a cybersecurity problem, calculate risks and support designing optimal cyber defenses dynamically.

Contents

List of Figures

List of Tables

List of Abbreviations

CRA	Cybersecurity Risk Assessment
CRM	Cybersecurity Risk Management
ISRA	Information Security Risk Assessment
ISRM	Information Security Risk Management
ISM	Information Security Management
BN	Bayesian Network
HBN	Hybrid Bayesian Network
ID	Influence Diagram
HID	Hybrid Influence Diagram
MAID	Multi-Agent Influence Diagram
FAIR	Factor Analysis of Information Risk
MC	Monte Carlo
EFBN	Extended FAIR-BN
ARA	Adversarial Risk Analysis
D-A	Defend-Attack
NVD	National Vulnerability Database
EV	Expected Value analyses
ALE	Annualized Loss Expectancy
LRAM	Livermore Risk Analysis Methodology
CRAMM	Central Computing and Telecommunications Agency Risk Analysis and Management Method
ISACA	Information Systems Audit and Control Association
OCTAVE	Operationally Critical Threat and Vulnerability Evaluation
COBIT	Control Objectives for Information and Related Technology
BAG	Bayesian Attack Graph
CEG	Causal Event Graph
DAG	Directed Acyclic Graph
CPT	Conditional Probability Table
JT	Junction Tree

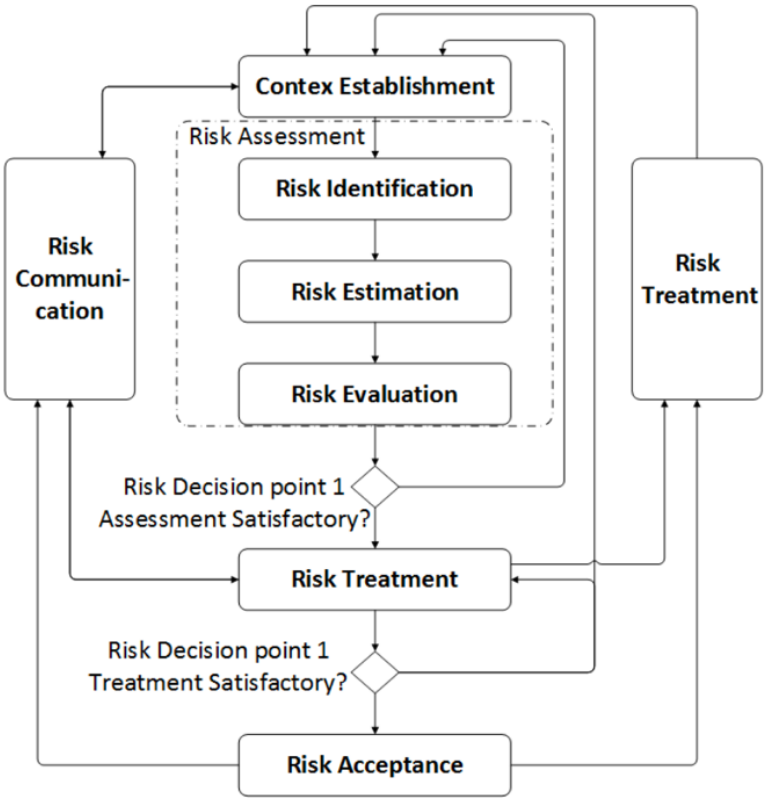
DD	Dynamic Discretization
MCMC	Markov chain Monte Carlo
MLE	Maximum Likelihood Estimates
DT	Decision Trees
C&C	Command and Control
BMD	Bounded Metalog Distributions
LEF	Loss Event Frequency
LM	Loss Magnitude
TEF	Threat Event Frequency
V	Vulnerability
CF	Contact Frequency
PoA	Probability of Action
TC	Threat Capability
RS	Resistance Strength
PL	Primary Loss
SL	Secondary Loss
SLEF	Secondary Loss Event Frequency
SLM	Secondary Loss Magnitude
RA	Risk Aggregation
IID	Independently Identically Distributed
CDF	Compound Density Factorization
MD	Metalog Distribution
DDoS	Distributed Denial of Service
gbps	gigabits per second

Chapter 1 Introduction

With the increasing penetration of Internet technology into every aspect of life, and the increased complexity of the networked environment, governments and business organizations aim to protect themselves against a diverse range of cyber risks. These risks, usually associated with data breaches, such as, e-mail leakages, and abnormal online operations, such as forced interruptions in online services, cause economic losses and other serious consequences [1]. Cybersecurity has become a matter of global interest and importance. For instance, as one of the world's leading digital nations, the UK lists cybersecurity as a top priority and the five-year budget for national cyber security program increased significantly from £860 million in 2011 to £1.9 billion in 2017 [2].

Best practice cybersecurity depends heavily on the risk management process [3] [4] [5], where Cybersecurity Risk Management (CRM) involves assessing and managing the risks of exposure to information systems that may be attacked, damaged or disrupted, as well as the task of securing the confidentiality, integrity, and availability of information in these systems [1] [3]. A well-recognized CRM process is the one provided by ISO/IEC 27005 [6], which encourages the practice of continuously identifying, reviewing, treating and monitoring risks to achieve risk acceptance. We illustrate this process in Figure 1-1.

Figure 1-1 Risk management process [6]



Cybersecurity Risk Assessment (CRA) and decision making is the foundation of CRM and is the topic of the thesis. Hubbard states that *"Putting cybersecurity risk assessment and decision-making methods on a solid foundation will affect everything else cybersecurity does."* in [1]. CRA aims to prevent and mitigate cyber risks by helping risk managers identify and prioritize risks, allocate resources to control or alleviate them, and make decisions about cyber defence strategy [4] [7].

CRA consists of three phases in general, which are risk identification, estimation, and evaluation. For an organization, the preliminary practice of CRA might involve these steps: 1) list information assets that are vulnerable to be attacked (risk identification); 2) qualify the risk (i.e., financial losses that the organization might bear) associated with each listed information asset being attacked (risk estimation); 3) prioritize risks against criteria for risk acceptance and objectives relevant to the organization as most organizations operate on a limited security budget (risk evaluation). Based on the CRA an optimal protection strategy regarding cyber

strategy and finances can be determined and hopefully achieve appropriate protection or risk reduction [8].

There are multiple approaches of CRA and many practices for conducting risk estimation [9] [10], but most CRA approaches agree that asset value, threat, and vulnerability are the key features that define information security risk [10]. An asset is defined as an information source of value to the organization. A threat is posed by an opponent (cyber attacker) that may be a position to act. Vulnerabilities are points of weakness in the system that a threat might exploit to gain access to the asset. Weaknesses may be intrinsic to a piece of software, such as a zero-day exploit, or introduced through misconfiguration, or human negligence. These weaknesses might then be mitigated or eliminated by the defences or risk controls. Therefore, vulnerability can be viewed as that product of the interplay between the defender and the attacker.

By quantitatively modelling these features of CRA, and their relationships, the risk assessor can identify adverse events, and produce a risk estimation with associated probabilities of occurrence and expected value of consequential losses. The decision maker then uses this estimation to determine whether a risk is acceptable or not. If a risk is deemed unacceptable the organization may consider implementing risk treatments, either by improved defences, involving better mitigation, avoidance, or transfer of the risk to another party, via insurance. In some cases, the risk itself may be tolerated, if the cost of improved defences is too high. Although there are numerous variations in the criteria used when making these decisions, rational and accurate risk estimation should always inform them.

1.1 Challenges of Cybersecurity Risk Modelling and Calculation

Managing cybersecurity risk is not an exact science. It brings together the collective judgments of individuals and groups that engaged in cybersecurity, drawn from across all operations within, or between, organizations. Towards this, numerous standards and approaches have been proposed (we introduce this in section 2.2).

Two mainstream directions of CRA are top-down organizational risk analysis and bottom-up technical risk analysis. The first is motivated by the need for organizational level decision making, which is primarily economic and budgetary. The second is driven by the perspective of cybersecurity analysts designing cyber defenses that consider attacker and defender interaction.

We can summarize the challenges of cybersecurity risk modelling and calculation and from the aforementioned directions:

1) Modelling complex cyber problems with explanatory risk estimation

The problem with many of the CRA standards and approaches is that they concentrate mainly on general principles and guidelines while leaving users without the depth needed for successful implementation [11]. Moreover, most of them fail to provide managers or cybersecurity analysts with a solution which can help to reveal cause-effect relationship between risk factors and furthermore generates an explanatory risk assessment that guides decision making.

We find that many of these standards and approaches have adopted the idea that risk estimation is simply the product of probability (or likelihood) of the risk and the impact (or loss). Both probability and impact are usually measured on a scale of, i.e., 0 to 1 and 1 to 10, and the resulting number is used to represent the size of the risk. The international standard, ISO/IEC 27005, can be an example for this kind of measurement. This type of risk measure is quite useful for prioritizing risks (the bigger the number, the greater the risk), but it is difficult to apply to concrete problems and can be irrational when applied blindly. Moreover, it is generally not sufficient for decision making. We believe that it is possible to avoid all these problems by using causality and probability to inform risk modelling and calculation.

2) Conducting CRA from an organizational perspective:

Conducting CRA at an organizational level is required for high level decision making, which is primarily economic and budgetary. The Factor

Analysis of Information Risk (FAIR) model provides a method to address this since it identifies a set of risk factors interact with each other to ultimately estimate cyber financial losses to an organization. However, the FAIR model has several limitations, including the inability to deal with long-tail risks, informally known as 'black swan' events, restricted assumptions for variable probability distributions and limited expandability, demonstrated by its very shallow approach to vulnerability modelling.

3) Conducting CRA from a technical perspective:

Conducting CRA from a technical perspective is required by cybersecurity analysts who design cyber defenses to defend against attackers. Game-theoretical approaches have been proposed as the solution here [12-15]. However, conventional game theory faces challenges when the problem and associated game models get more realistic and complex, as it requires the calculation of a Nash equilibrium for the game problem, and it assumes that common knowledge is hold by game adversaries. Both assumptions are unrealistic in the cybersecurity context. Adversarial Risk Analysis (ARA) is a method proposed to address these shortcomings. However, most ARA solutions use Monte Carlo (MC) simulation to carry out the calculations, which is straightforward to implement but can become computationally challenging when dealing with complex decision dependent uncertainties, i.e., a game with longer decision sequences. Moreover, it cannot cope with new evidence that could be used to update the game model, dynamically, in real time, which we contend is a realistic requirement for practical use.

1.2 Research Objectives

To address these challenges a number of research objectives were identified:

1) **Modelling complex cyber problems with explanatory risk estimation:**

   Objective I: Introduce causal analysis and related Bayesian network technology to underpin cybersecurity risk modelling and provide an example using the kill chain model.

2) **Conducting CRA from an organizational perspective:**

   Objective II: Reveal the limitations of the FAIR model.

   Objective III: Create a BN alternative to the FAIR model which eliminates FAIR's restrictions and delivers improved practical utility.

   Objective IV: Evaluate the accuracy of the FAIR and FAIR-BN approaches and identify the pros and cons in both approaches.

3) **Conducting CRA from a technical perspective:**

   Objective V: Propose an HBN-based alternative framework for ARA and identify its advantages compared with the state of the art.

   Objective VI: Apply the proposed framework to solve more practical D-A problems, i.e., involving extra risk variables and longer decision sequence.

   Objective VII: Provide a mechanism in the framework to support dynamic decision making in multi-period D-A games and present working examples.

1.3 Associated Publications

Publication 1:

Jiali Wang, Martin Neil, and Norman Fenton. "A Bayesian network approach for cybersecurity risk assessment implementing and extending the FAIR model." *Computers & Security* 89 (2020): 101659.

Publication 2:

Jiali Wang and Martin Neil. “A Bayesian-network-based cybersecurity adversarial risk analysis framework with numerical examples.” arXiv:2106.00471

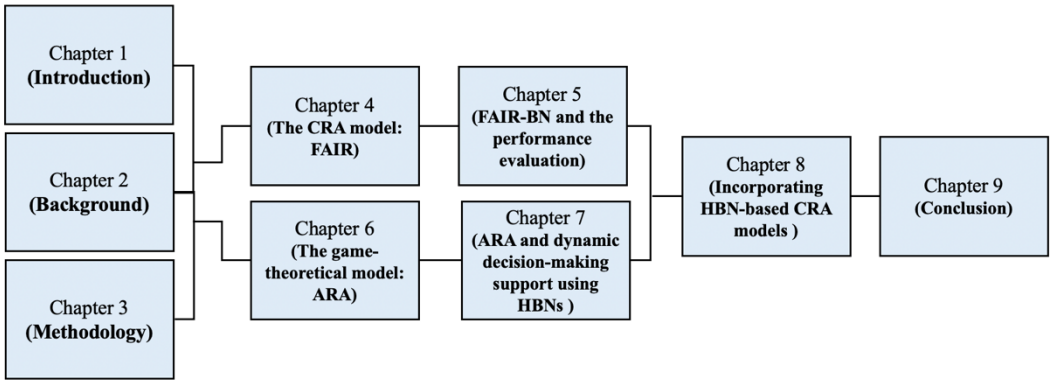
Publication 3:

Jiali Wang and Martin Neil. “Cybersecurity adversarial risk analysis using hybrid Bayesian networks.” (Submitted to International Journal of Operational Research)

1.4 Thesis Structure

The thesis is structured as shown in Figure 1-2.

Figure 1-2 Thesis structure



In chapter 1, we provide an outline of CRA, summarizing challenges in modelling and calculation in cyber problems and accordingly specify the objectives of this thesis.

In chapter 2, we provide an outline of how CRA can be organized and review the risk assessment standards and approaches in cybersecurity. We claim that it is possible to avoid limitations of many of these standards and approaches by adopting causality and probability. We introduce causal analysis which is implemented using Bayesian networks and review the application of Bayesian networks in cybersecurity. This chapter sets the context of our research represented by this thesis.

In chapter 3, we introduce Bayesian networks including the background knowledge, the related technology that are employed in this thesis and examples of BNs in cybersecurity risk assessment. Specifically, we use a kill chain model to illustrate how causal analysis can be applied for cybersecurity risk modelling. This chapter achieves the objective I.

In chapter 4, we provide an introduction of Factor Analysis of Information Risk (FAIR) and its position in CRM. Moreover, we reveal several important limitations that FAIR has, based on detailed analysis of the assumptions of the FAIR model,

focusing on its taxonomic structure and algorithms and provide an in-depth analysis of the calculation mechanism of the FAIR model, which has hitherto not appeared in the literature. This chapter contributes to objective II and is covered by our Publication 1.

In chapter 5, we describe how we construct FAIR-BN to faithfully represent the FAIR model and illustrate the flexibility of FAIR-BN. We introduce a means for evaluate the performance of FAIR and FAIR-BN. Experiments comparing the FAIR and the FAIR-BN models are provided and analysed. This chapter contributes to objectives III and IV and is covered by our Publication 1.

In chapter 6, we introduce Adversarial Risk Analysis (ARA) and illustrate the calculation mechanism of it focusing on a typical game model, sequential Defend-Attack (D-A) models, for this kind of models can properly represent realistic cybersecurity cases. We propose an alternative framework, based on hybrid BN inference and decision trees, to solve the D-A games from the ARA perspective. This chapter contributes to objective V and is mainly covered by our Publication 2.

In chapter 7, we illustrate how to use the proposed framework introduced in Chapter 6 to solve more practical D-A problems involving extra variables and longer decision sequences. Also, we present numerical examples to show how our framework can support decision making in different application contexts involving extra variables, longer decision sequences and dynamic decision making. This chapter contributes to objectives VI and VII and is mainly covered by our Publication 3.

In Chapter 8, we incorporate a process-oriented model and a D-A model, with the FAIR-BN, to show the expandability of the proposed framework and explore the potential to expand the risk modelling of the FAIR-BN. This chapter contributes to objectives III and is mainly covered by our Publication 1.

In Chapter 9, we conclude the work of the thesis with specifying how the objectives summarized in section 1.2 are overcome/alleviated by our work.

Chapter 2 Cybersecurity and Causal Risk Analysis

Identifying components of 'risk' and modelling relationships between the components for deriving the estimation of 'risk' constitutes the common theme in cybersecurity risk analysis. However, limitations exist in traditional statistics and data analysis methods for risk assessment, from the way they define 'risk', identify the components of 'risk' and how they organize the estimation of 'risk'. We claim that analyzing cybersecurity risk from a causal perspective is possible to avoid all these problems and ambiguities surrounding the term 'risk'.

In this chapter, we firstly present an overview of Cybersecurity Risk Assessment (CRA) in section 2.1 by introducing how it is generally organized. We review CRA standards and approaches in section 2.2 and argue that the key ingredients to inform risk which are causality and probability, are missing from these standards and approaches. We formally introduce causal analysis, which can be implemented using Bayesian networks in section 2.3. In section 2.4, we review the current applications of Bayesian networks in cybersecurity. Most of these works focus on certain phase of CRA, i.e., risk identification, estimation and evaluation or the defence decision making individually, while lack of producing an integrated CRA and decision-making solution, which is achieved in our work. All these analyses set the context of our research represented by this thesis.

2.1 An Outline of Cybersecurity Risk Assessment

The Information Security Risk Management (ISRM) process is an iterative process of reviewing and monitoring risks structured by the international standard ISO/IEC 27005 [6] as is illustrated in Figure 1-1. This standard has also declared three phases for risk assessment, which are risk identification, risk estimation and risk evaluation. In this section, we introduce how CRA might be organized from these three phases.

2.1.1 Risk Identification

Most CRA approaches agree that asset value, the vulnerabilities of an asset and the threats that might take advantage of these vulnerabilities are the key features that define information security risk [10]. Risk identification, therefore, generally revolves around these three features.

A critical first step in typical risk analysis is identifying and valuating information assets/resources. In general, information assets are assets which are valuable to the organization and are either tangible or intangible [16]. ISO/IEC 27001 [3] requires organizations to generate and maintain an up-to-date inventory of information assets as the basic for cybersecurity risk management. Examples of information assets include business processes, information, computer hardware and software, network, personnel, documents, and the organization's reputation [6]. These assets can be valuated based on their importance to the organization [17]. Landoll [16] defines three quantitative asset valuation appraisements which are: cost valuation, market valuation and income valuation. He also defines four types of qualitative asset valuation appraisement including Binary, classification-based, rank-based and consensus-based asset valuation.

From a relatively high-level view, vulnerabilities can be identified using the following methods: 1) automated vulnerability scanning tool, 2) security testing and evaluation, 3) penetration testing and 4) code review [6] [17]. Since these methods can yield some "false positives", there are activities may be considered in practice as the supplement, which are: 1) on-site interviews; 2) questionnaires; 3) physical inspection and 4) document review [6] [9]. Vulnerabilities can also be identified from the results of previous risk analysis, IT system audit reports, system anomaly reports, security review reports, and system test and evaluation reports [6]. Other potential sources are public vulnerability databases, like the National Vulnerability Database (NVD) [18].

Threats are identified and documented through a formal process called threat modelling [9]. Several methods available to identify threats to a system and map them with the related vulnerabilities are described in [19]. ISO/IEC 7498-2 defines

a reference model of major security threats, of which the main perspectives for identifying threats include: 1) destruction of information and/or other resources; 2) corruption or modification of information; 3) theft, removal, or loss of information and/or other resources; 4) disclosure of information; 5) Interruption of services [20]. This can be used as a checklist for threat identification.

2.1.2 Risk Estimation

In the risk estimation step, the organization needs to choose a model to measure the risk. Risk model specifies the relationship between risk factors which include assets' value, vulnerability effect, threat impact, threat likelihood, and so on [21]. Based on the chosen model, the risk value for each incident scenario can be measured. Cybersecurity risk estimation methods are generally classified into three categories that are quantitative, qualitative and hybrid (also called semi-quantitative).

Quantitative risk estimation relies on certain numbers, time-consuming calculations, statistics and probability theory to identify the level of an organization's risk exposure [22]. Quantitative risk assessment is generally based on objective measurements, and the results can be expressed in a management-specific language (i.e., financial value, percentages, and probabilities) [23]. The inputs and outputs of quantitative risk assessment can be classified in two categories: *financial* and *non-financial*. In financial evaluation, a monetary value is assigned to every asset, threat, vulnerability and security defence deployment. In contrast, non-financial evaluation yields a non-financial number, i.e., the occurrence probability of a cyber accident.

Quantitative CRA methods are also known as "expected value analyses" (EV) [24]. In these methods, the expected loss to the organization caused by each scenario is used to valuate the impact of the scenario. The calculated impact, together with the probability of the threat, forms a quantitative risk estimation of a particular scenario. The best known EV appraisements is Annualized Loss Expectancy (ALE).

The major problem with the quantitative approach is the high-technique-required and time-consuming process, which depends on the detailed information and calculation mechanism. Information such as the value of the assets and the historical

incident data is used to calculate the expected loss and determine the probability. Due to the limited time, money, and human resources available for organizations, implementing this approach will not be a trivial task [20] [22].

Many organizations find that qualitative information security risk assessment is sufficient for their preliminary cybersecurity risk management requirement [16]. In a qualitative CRA approach, ranks and relative values are used to represent the impact and possibility of a particular information risk scenario. Information security risks are estimated using methods and principles with qualitative levels. Generally, the input and output of qualitative risk assessment can be classified into two categories: *range variables* and *linguistic variables* for input, and *range variables* and *rank variables* for output [9]. For range variables, which can be both input and output, they are usually represented by numerical ranges, i.e., 0 to 5. Rank variables are usually expressed on a on a scale of three to five levels (e.g. low, medium, and high) [17]. For linguistic variables, expert opinion is used to determine states of variables, which is useful for handling situations that are not well defined [25].

These qualitative approaches are widely used for preliminary risk analysis, because there is often insufficient accurate historical data to calculate the impact and probability of risk event occurrence, and also because they are much easier to understand and implement [17]. Also, with these approaches, the assessment of assets, threats, and vulnerabilities is more straightforward and meanwhile the calculations involved are simpler [21]. However, the lack of sufficient measurable detail to support cost-effective decision making for management is the main restriction of these approaches [22]. Moreover, the qualitative expression of knowledge and experience of assessors makes the measurement more subjective, hard to compare and imprecise than their quantitative counterparts [17]. Another limitation with the traditional qualitative appraisals is that the range of values assigned to information assets, their vulnerability level, and the threat likelihood is comparatively small, which makes it difficult to prioritize information security risks and compare the associated risk assessment results [21].

Because of the strengths and weaknesses of both the quantitative methods and qualitative methods, several hybrid methods were proposed. Among these methods, Bayesian probability related methods have received more and more attention due to their effective combination of objective data and subjected data (i.e., personal belief represented by probability) and producing explanatory quantitative risk estimation results.

2.1.3 Risk Evaluation

Risk evaluation is the process of rating risks on a scale and against accepted risk criteria to determine the significance of each risk [3]. Appropriate steps are required to be determined for managing risks and addressing them properly [17]. In this phase, the identified risks need to be prioritized based on their legal, financial, or reputational impact to the organization and their relative probability of occurrence in order to make defence decisions [26].

There are four ways to address a particular risk as summarized in [9], which are: 1) accept, meaning the organization understands the risk and its consequences and consciously decides to accept it; 2) avoid, meaning the activity that is exposing the organization to one or more risks is avoided altogether; 3) transfer, meaning that all or part of the responsibilities and liabilities associated with a particular activity and the related risk are shifted to another party and 4) mitigate, meaning the risk and its consequences are controlled and limited in some way, reducing the risk to a level that is lower than the organization's acceptance level.

In general, handling an organization's risks involves a combination of these ways: some of them are avoided, some are transferred, some are mitigated, and the rest are accepted [26]. As a part of the risk evaluation process, the risk acceptance level (criteria) should be determined depending on the organization's goals, objectives, policies, and the interests of stakeholders. All the risks under this level can be accepted or tolerated by the organization's management [6].

2.2 Risk Assessment Methods in Cybersecurity

In industrial practice and academic research, numerous works have been developed towards Cybersecurity Risk Assessment (CRA) and decision-making support which are the core of Cybersecurity Risk Management (CRM). These works differ from the concerned cyber risk objects (i.e., information asset or related systems [27]), definition of risk (i.e., financial or non-financial), and the phases of CRA which are risk identification methods (i.e., automatic cyber threat detection or expert-knowledge-based risk identification), risk estimation methods (i.e., qualitative, quantitative or the hybrid methods) and risk estimation criteria (i.e., bias between security level and financial costs).

Several CRA standards and approaches, which consider the entire three steps of CRA, have been developed. These approaches include: ISO/IEC 27005 [6]; NIST SP 800-30 [21]; Factor Analysis of Information Risk (FAIR) [28] [29]; Central Computing and Telecommunications Agency Risk Analysis and Management Method (CRAMM) [30]; CORAS [31]; Information Systems Audit and Control Association (ISACA) [32] and the Operationally Critical Threat and Vulnerability Evaluation (OCTAVE) [33]. These standards and approaches are detailed reviewed in several works [26] [11] [34]. Although, to conduct preliminary CRM, all of these CRA approaches can be chosen by organizations [35], unneglectable limitations still exist.

It is claimed that the problem with many of these methodologies is that they concentrate mainly on general principles and guidelines, leaving users without adequate details for implementation [11]. It is also argued by Ekelhart in [34] that even industry standards, like COBIT (Control Objectives for Information and Related Technology) [36] and ISO/IEC 27005 fail to provide managers with a clear and simple visualization of the security risk assessment.

We have also realized these limitations, since we find that most of these methods have applied the idea of standard impact-based risk measurement, i.e., risk matrix (or its variation, heat map) for risk valuation. In the standard impact-based risk measurement, risk is valuated based on the possibility (or likelihood) of the risk and

the impact (or loss) the risk can cause. Typically, risk analysts measure both probability and impact on a scale of, i.e., 1 to 5 or 1 to 10, and use the resulting number representing the size of the risk. Risk matrix can be regarded as an extension of the standard impact-based risk measurement. We borrow the risk matrix used in ISO/IEC 27005 as an example to illustrate how it works. This risk matrix is shown in Figure 2-1. Based on it, risk is mapped to an overall risk ranking, for example, score 0-2 represents low risk, score 3-5 represents medium risk and score 6-8 represents high risk [6].

Figure 2-1 The risk matrix used for CRA in ISO/IEC 27005 [6]

	Likelihood of incident scenario	Very Low (Very Unlikely)	Low (Unlikely)	Medium (Possible)	High (Likely)	Very High (Frequent)
Business Impact	Very Low	0	1	2	3	4
	Low	1	2	3	4	5
	Medium	2	3	4	5	6
	High	3	4	5	6	7
	Very High	4	5	6	7	8

Given the difficulty of quantifying risk, and the lack of time usually given over to addressing it thoroughly, the probability and impact numbers needed on an impact-based risk measurement are often replaced by labels (like low, medium, high) and the measurement results are plotted on a heat map which uses different shades of colour to represent the severity of a risk.

This type of risk measure is quite useful for prioritizing risks (i.e., the bigger the number, the greater the risk), but it is normally impractical and can be irrational when applied blindly. Moreover, it is generally not sufficient for decision making. We believe that it is possible to avoid all these problems by using the causal analysis [37]. We introduce causal analysis and explain how these limitations can be avoided in section 2.3.

Except for the mentioned standards and approaches which estimate cybersecurity risk based on impact-based risk measurement and risk matrix, several process-oriented methods are proposed, which provide graphical notations to illustrate the

attacker's goals with possible routes to reach these goals and accordingly estimate risk as the probability of these goals are achieved. These methods, which initially reflect the idea of causal analysis, provide an explanatory and more meaningful risk estimation compared with the risk matrix methods, since the events and the relationships between them are explicit meanwhile probability values which represent the likelihood of each event are used to produce the calculation.

Typical process-oriented risk estimation paradigms include threat trees [38], attack trees [39], attack graphs [40], defence trees [41] and intrusion kill chains [42]. Attack trees are methodology which decomposes the attack events to series of pre-conditional events as a tree structure [43, 44]. They are multi-level diagrams consisting of one root, and several level of leaves. The leave nodes are conditions which must be satisfied to make the direct parent node to be true; when the root is satisfied, the attack is complete. This kind of risk estimation models can be used to prioritize risks in an information system and to evaluate security controls for known vulnerabilities [39]. The defence tree is an extension of attack trees with added leaves representing controllable countermeasures. However, a majority of these tree models fail to consider the attacker's capabilities and, consequently, the likelihood of a particular attack being executed [45]. To alleviate such drawbacks, Bayesian Attack Graphs (BAG) [45, 46] and the security graph model [47] have been proposed as alternatives. These approaches apply Bayesian probabilistic logic to conduct CRA. Intrusion kill chains are also a kind of popular process-oriented model. They are designed to model attack processes, with the aim of highlighting patterns within individual intrusions and how they may fit into part of a larger threat. Enough attacks need to be collected and modelled and after that common attack steps could be established. In other word, this proposal can be regarded as building a library of attack steps [42]. Intrusion kill chains can also be interpreted using Bayesian probabilistic logic following the idea of causal analysis to achieve more flexibale modelling and high-efficient calculation. We introduce causal analysis in the next section. Since causal analysis is often implemented as Bayesian networks, we introduce Bayesian network technology for causal modelling, probobilistic calculation and decision analysis in detail in Chapter 3. An example of interpreting

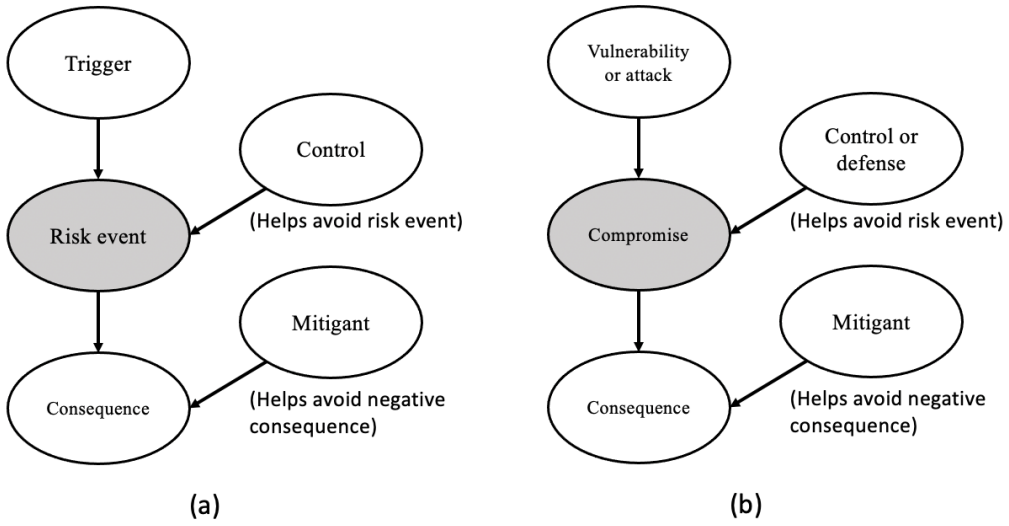
intrusion kill chain using Bayesian networks is also provided in Chapter 3 after the related technology are introduced.

2.3 Causal Risk Analysis

The idea of causal risk analysis is explained in [37]. With the provided causal perspective of risk, a risk is characterized not by a single event but by a set of events in a causal chain. We introduce characteristics of the causal model and illustrate how it can lead to an explanatory risk modelling with avoiding limitations that the impact-based risk measurement has in this section.

In the causal context, both risks and opportunities are considered. The key concept of causal risk analysis is that a risk (and, similarly, an opportunity) is an event that can be characterized by a causal chain involving five components which are: 1) the event itself; 2) at least one consequence event that characterizes the impact (so this will be something negative for a risk event and positive for an opportunity event); 3) one or more trigger (i.e., initiating) events; 4) one or more control events that may stop the trigger event from causing the risk event (for risk) or impediment events (for opportunity); 5) one or more mitigating events that help avoid the consequence event (for risk) or impediment event (for opportunity) [37]. In the cybersecurity context, we are more interested in the causal view of risk, since in most scenarios the mainly concern is to conduct defence deployment to avoid possible losses rather than looking for opportunities that can generate positive returns. We show the causal taxonomy of risk and its instantiation in cybersecurity in Figure 2-2.

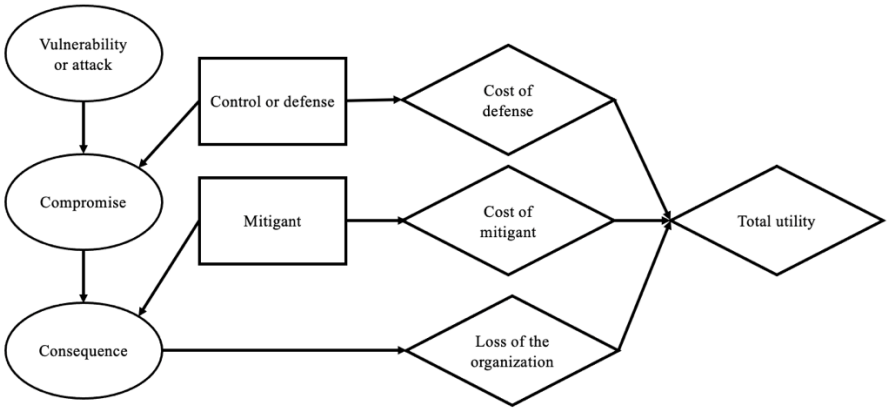
Figure 2-2 Causal taxonomy of risk. (a) causal view of risk. (b) an instantiated causal risk model in cybersecurity



These events each have several possible outcomes (Boolean, Labelled or numerical values). The uncertainty associated with a risk can not only be a separate notion (as low, medium, and high assumed in the classic approaches). Every event (and hence every object associated with risk) has uncertainty that is characterized by the event's probability distribution.

Although all the events in the causal model shown in Figure 2-2 are treated as uncertain variables, there are clear differences between them from a "decision theory" perspective. Both the control node and the mitigant node represent decisions that we can consider performing or not. In contrast, the other nodes are chance nodes. Some important components which are missing from the model are utilities. In general, any decision (i.e., deploy a defence or not) will have a cost, which we can think of as a negative utility, and every consequence node (i.e., the loss caused by a cyber attack faced by the organization) will also has a cost. The whole picture of the causal model therefore can be illustrated by the one shown in Figure 2-3, where decision nodes are represented as rectangles and utility nodes as diamonds. This kind of models is known as influence diagram, which is generally implemented using Bayesian networks and can be used for decision analysis. We provided detailed explanation of influence diagram in section 3.4.

Figure 2-3 Influence diagram version of the causal model in cybersecurity



A causal analysis is beneficial because once a risk event is identified from a particular perspective, i.e., the defender of information systems, there will be little ambiguity about the definition of *risk* and a clear causal structure that helps tell the story behind the risk scenario being analysed.

The pioneering research by Judea Pearl on Bayesian networks has provided the philosophical and practical ideas on how to elicit, articulate and manipulate causal models [48] [49]. Likewise, the work on causal idioms and the dynamic discretization algorithm has been widely applied to make efficient modelling and accurate and productive calculation in HBNs [37] [50]. Also, there are software products, such as AgenaRisk, which implement the underpinning algorithms needed to do causal and probabilistic modelling and furthermore help to implement and calculate Bayesian network models intuitively and provide insightful results to decision makers even for complex problems. We introduce the related technology in Chapter 3.

2.4 Application of Bayesian Networks in Cybersecurity

Bayesian Networks (BNs) can explicitly describe casual and probabilistic relationships between variables, using a qualitative structure, based on conditional probability, that, along with inference algorithms, can be used to represent and solve risk assessment and decision making problems under uncertainty [51]. Also, BNs can incorporate objective and subjective information and support bidirectional inferences to reason from cause to effect and vice versa [52]. The computations in BNs are based on the Bayes' theorem and this provides the only rational and consistent way to update a belief in an uncertain event when we observe new evidence related to that event [37]. These benefits of BNs have made them a powerful tool for risk assessment and decision analysis [53]. Bayesian Networks become increasingly popular in the cybersecurity field while a plenty of related works are published in recent years [54]. In general, Bayesian networks are used to identify vulnerabilities of, and the threats to information assets or systems, predict/diagnose the associated risks and evaluate the effectiveness of protect strategies [24-27] [18, 28-32].

Bayesian networks were used to automaticaly categorize vulnerabilities according to their security types by Wang and Guo [55]. A BN model developed in [56] is used to identify compromised users in shared computing infrastructures based on alerts. Insider threats were predicted by a Bayesian network in [57]. Probability values of certain events in cybersecurity are estimated by work [58] and [59]. A BN-based model was constructied to predict the probability of a data breach in a bank caused by a malicious insider [58]. A cyber security risk estimation result which can be used as a firm's security profile and data breach statistics was generated using Bayesian networks in [59]. Dynamic Bayesian networks are specifically applied for conducting risk estimation in dynamic domains which mainly utilizes the characteristic of BNs wherby the degree of belief is updated when new observations are made available. For instance, a dynamic Bayesian network is proposed to update the states of an information system and predict further status of the system in [60]. Frigault proposes a dynamic Bayesian network model to tackle with temporal risk factors faced by a network in different time

phases and continuously measure the network security in a dynamic environment [61].

Some process-oriented models implemented using BNs have shown the idea of causal analysis. Bayesian attack graph was introduced to model potential attack paths by Liu and Man [62]. They use BNs to enhance the conventional attack tree model by considering the attacker's capabilities and the likelihood of a particular attack being executed. In addition, they develop algorithms to compute an optimal subset of attack paths based on background knowledge of attackers and attack mechanisms. Based on their work, Bayesian attack graphs are proposed to quantify the chances of network compromise at various levels by Poolsappasit et al [45]. Moreover, Pan [63] proposes the use of Causal Event Graphs (CEGs), which are designed to model the causal relationship between devices in a cyber-physical model to provide risk estimation.

BNs are now routinely used for prediction and diagnostic inference tasks and have been extended, in the form of Influence Diagrams (IDs) [64], to support decision making tasks. Khosravi-Farmad and Rezaee use Bayesian decision networks in risk mitigation by modelling dependence between vulnerabilities of the system and the threats faced by the system to find appropreate security contermeasures considering costs of implementation [65]. For cases with more than one decision makers, Multi-Agent Influence Diagrams (MAIDs) are proposed as an extension of IDs [66].

However, to the best of our knowledgy, none of the existing works on using Bayesian network technology in cybersecurity risk have provide an integrated framework which can handle causal risk analysis and decision analysis as a whole (as illustrated by the infuence diagram version of causal model in Figure 2-3) and consider the two mainstream directions of CRA which are the top-down organizational risk analysis and bottom-up technical risk analysis. We provide this framework in our research represented by this thesis.

Chapter 3 Bayesian Networks and Models of Cybersecurity Risk

This chapter introduces the fundamental technology, Bayesian Networks (BNs), adopted in this thesis. BNs can be used for modelling causal and probabilistic relationship between variables and to conduct the related calculation for risk assessment and decision analysis and are especially applied in cybersecurity problems in this thesis. We provide an overview of Bayes' theorem and Bayesian networks in section 3.1. In this thesis, we build BNs, including constructing the network structure and clarifying the relationship between variables using expert's knowledge. How the BNs can be constructed based on experts' knowledge is introduced in Section 3.2. The development of probabilistic inference algorithms, introduced in section 3.3, allows the use of Hybrid Bayesian Networks (HBNs), which can model both continuous and discrete variables, with high calculation efficiency. A kind of BNs, influence diagrams, which are developed for decision analysis are introduced in Section 3.4. Section 3.5 uses examples, i.e., a kill chain model, to illustrate how HBNs can be used for causal modelling in cybersecurity context.

3.1 From Bayes' Theorem to Bayesian Networks

Bayesian probability was initially proposed by Thomas Bayes in the 18th century. It provides a way to reason coherently about uncertainty. Bayes' theorem enables us to update a prior probability for some unknown event when we see evidence about the event [37]. Equation 3.1 represents the Bayes' theorem, where H stands for our prior hypothesis of uncertainty towards some event and E is new observable evidence that related to the event. Rely on Bayes' theorem, if we observe a certain result of evidence E, our belief of H can be updated, which is $P(H|E)$ meaning the probability of H given E. This is also called the posterior probability of the hypothesis. In the cybersecurity context, for instance, we can use Bayes' theorem to update our belief of an information asset being attacked by taking the observed

evidence, i.e., the attack capability or the defence level, into consideration. Related examples are provided in Section 3.5.

$$P(H|E) = \frac{P(H)P(E|H)}{P(E)} \tag{3.1}$$

There are usually many unknown events and different pieces of evidence, and we can represent such problems graphically, using nodes to represent the uncertain variables and an edge between two nodes to represent their conditioning relationship. By this way, we can obtain a Bayesian network (BN) [37]. Bayes' theorem is applied to the calculation through a BN to correctly update evidence in a more complex problem.

Formally, a BN is a Directed Acyclic Graph (DAG) representing a joint probability distribution. It consists of the qualitative and quantitative parts. The qualitative part includes nodes representing variables and arcs representing causal or probabilistic relationships pointing from parent variables to their child variable. The quantitative part is represented by the "probabilistic weight" between parent nodes and their child node, which can be modelled using Conditional Probability Tables (CPTs) [37], statistical and conditionally deterministic functions. The "probabilistic weight" of a node without parents is modelled by the marginal distribution of the node.

More precisely, for variables such as Boolean variables (whose states are true and false) and labelled variables (whose set of states is simply a set of labels), the probabilistic weights are represented by CPTs. For example, the CPT for a variable X_i contains a series of conditional probabilities which can be represented as $P\big((X_i|\, pa(X_i)\big)$ given states of its parent variables, $pa(X_i)$. For a node X_i without parents, the CPT is the marginal probability distribution of X_i, $P(X_i)$. For numerical variables, which could be discrete (such as the count of times that an information system might been attacked within a month) or continuous (such as the financial losses that an organization might encounter), the probabilistic weights can be modelled using conditionally deterministic functions and statistical distributions. Here, conditionally deterministic functions are simply mathematical functions such as $Z = X + Y$. This kind of function would be used if a node (i.e., Z) is valued as

the deterministic function of its parent nodes (i.e., X and Y). Statistical distributions are special functions that depict the conditional relationship between one child variable and its parent variables by specifying mathematical parameters that define the shape and scale and other properties of the child variable using its parent variables. For example, a child variable can be modelled by a Normal distribution with parameters for mean and variance come from its parent variables [37].

The conditional-independent relationship among variables, represented by the absence of arcs, allows simplification of a BN's joint probability distribution. Furthermore, the marginal distribution of the child variable can be obtained by marginalizing over its parent variables in the joint distribution [37]. To construct a Bayesian network model, including creating the DAG structure and specifying the involved CPTs, statistical or deterministic functions, expert knowledge, empirical data, or the combination of both can be adopted.

For example, we consider a simple BN which consists of three nodes. In this BN, node A and node B are parents of node C. The involved 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 general, the joint distribution of a BN can be calculated following formula (3.2):

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P\big((X_i|\, pa(X_i)\big) \tag{3.2}$$

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 [37]. Here we use a toy example, flipping coins, to illustrate how the DAG and CPTs can be set based on personal knowledge. We use node A and node B to represent the events of flipping two coins respectively in Figure 3-1. Nodes A and B together determine node C, which represents the event that two heads appear. The DAG and CPTs for involved variables can be created based on common knowledge (assuming the two coins are fair), which is shown in

CPTs in Figure 3-1. As is illustrated, the prior belief of event C is represented by its calculated marginal distribution. The probability of two heads appear is 0.25 and 0.75 for otherwise. If we observe new evidence, i.e., coin A shows a head, we can update the belief of the C. The posterior probability of node C, given the evidence that coin A shows a head, is illustrated in Figure 3-2.

Figure 3-1 A toy BN model: Flipping coins

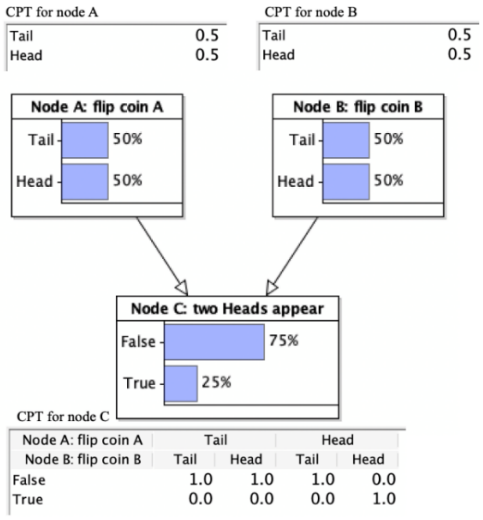
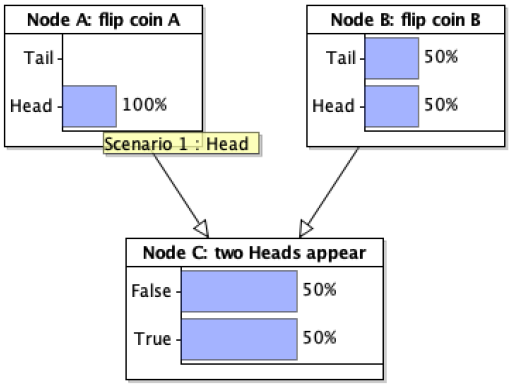


Figure 3-2 A toy BN model: Flipping coins-evidence entered



Junction Tree (JT) algorithm [67] is one of the typical algorithms for conducting high-efficient probabilistic inference in BNs with discrete variables. BNs which contain mixtures of continuous and discrete variables are called Hybrid Bayesian Networks (HBNs). For HBNs, the Dynamic Discretization (DD) algorithm [37] can be adopted to discretize the state range of continuous variables dynamically

according to the density of the distribution [50]. Then the HBN with discretized nodes are able to adopt the JT algorithm for high-efficient calculation.

These algorithms, including JT and DD, for conducting calculation in HBNs have been implemented and packaged in AgenaRisk [68]. AgenaRisk is a commercial BN software application which contains off-the-shelf functions for performing inference in HBNs, influence diagrams [37] (introduced in Section 3.4) and for performing compound sum calculations [67] (introduced in Section 5.2). We have used AgenaRisk for constructing and calculating HBN models in this thesis.

BNs have multiple of benefits summarized in [37], which include: 1) the ability to explicitly model causal relationship between risk factors, 2) can support bidirectional inference, meaning we can reason from effect to cause and vice versa, based on a rigorous probabilistic foundation, 3) can reduce the burden of parameter acquisition, 4) are able to overturn previous beliefs in the light of new evidence 5) can make predictions with incomplete data whilst combining diverse types of evidence including both subjective beliefs and objective data and 6) can arrive at decisions based on visible, auditable reasoning. This range of benefits, together with the explicit quantification of uncertainty and ability to communicate arguments easily and effectively, makes BNs a powerful technology for handling causal analysis, risk assessment and decision making. They have prevalent applicability, including causal reasoning [69], diagnostic inference [64] and enabling statistical reasoning such as machine learning from data [70, 71].

3.2 Constructing Bayesian Networks using Experts' Knowledge

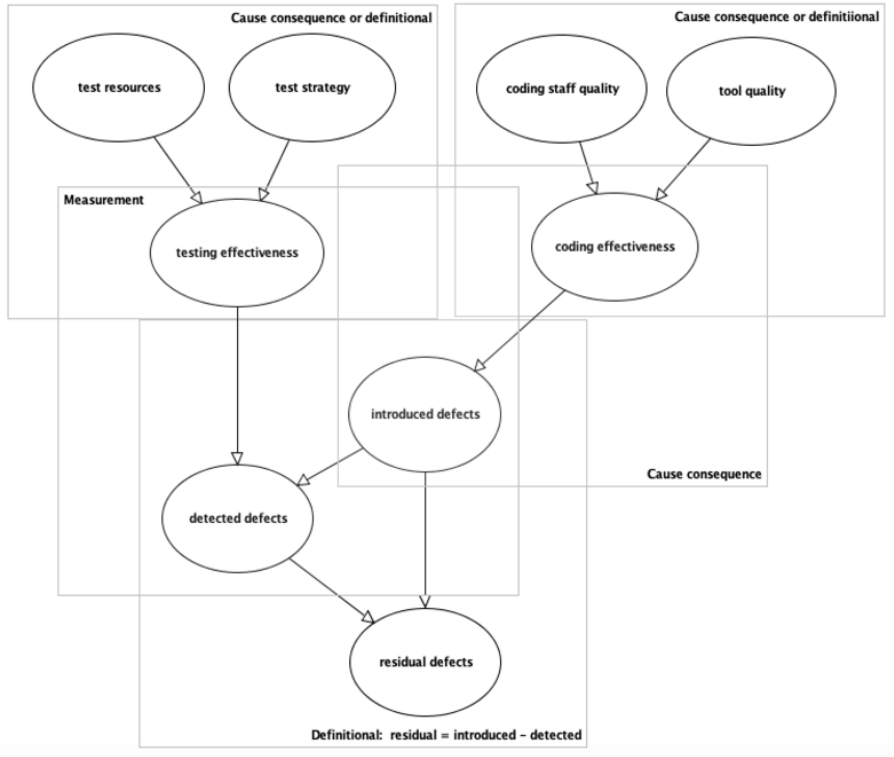
A BN can be built by specifying two components: a network structure which consists of nodes and directed arcs connected them, and the (conditional) probabilistic weight for each node. Both can be elicited from experts' knowledge and learned from data. In this thesis, we build BNs, including constructing the network structure and clarifying the relationship between variables (parameters) using experts' knowledge. We introduce how to construct BNs using experts' knowledge in this section.

Inspired by Pearl's idea [48] of assembling casual structures from a stock of building blocks, the work in [72] develops so-called idioms to serve as building blocks in BNs. These idioms represent common types of uncertain reasoning that can be used to construct BNs based on specific cases. They are recurring patterns that provide efficient and consistent guideline to reveal the casual relationship between variables and build the BN for similar problems.

Four especially common causal idioms are defined in [73], which are: 1) cause consequence idiom: the idiom models the uncertainty of a causal process with observable consequences; 2) measurement idiom: the idiom models the uncertainty about the accuracy of any type of measurement; 3) definitional/synthesis idiom: the idiom models the synthesis or combination of many nodes into one node for the purpose of organizing the BN. Also models the deterministic or uncertain definitions between variables; 4) induction idiom: the idiom models the uncertainty related to inductive reasoning based on populations of similar or exchangeable members. Here we use an example to illustrate how the idioms might be used to model a situation.

We consider a situation of testing defects of a software which is discussed in [72]. It is assumed that several defects can be introduced during the coding when develop a software product. The amount of the defects is determined by the coding effectiveness which can be influenced by factors, i.e., the coding staff quality and the tool quality. Testing the software results in discovering some/all the defects. It is represented by testing effectiveness which might be influenced by factors, i.e., test resources and test strategy. We call the defects that are not discovered as residual defects. We illustrate the relationship between involved variables of this situation in Figure 3-3. The related idioms are marked with rectangles.

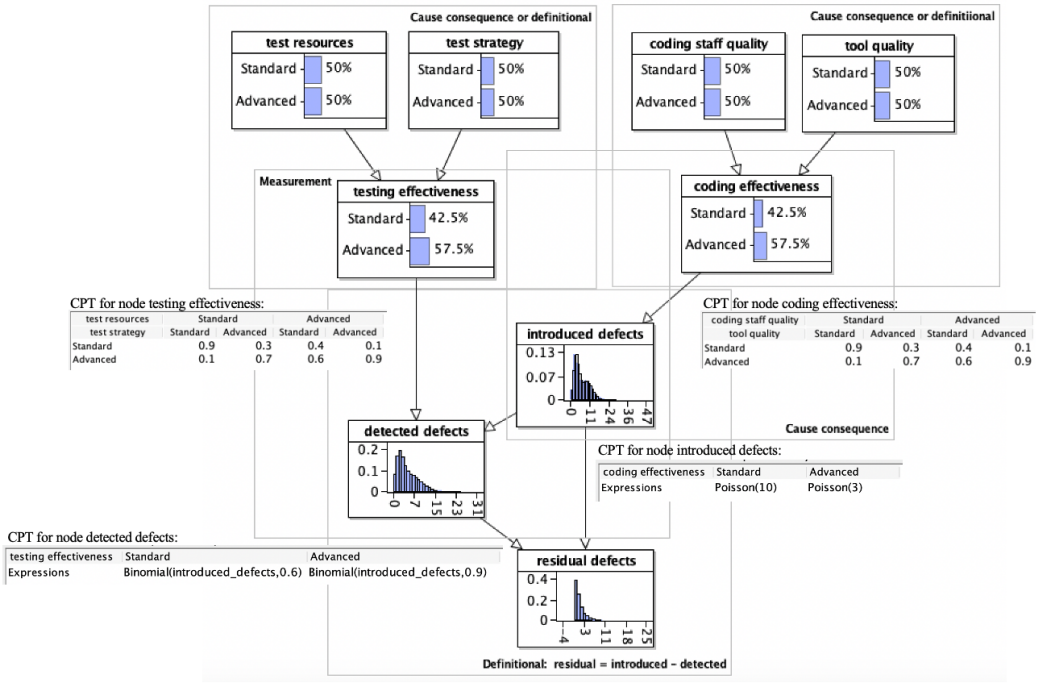
Figure 3-3 Idioms in the structure of a BN [72]



For idioms we are mainly interested in the graphical structure and for this reason, an idiom is not a BN but simply the graphical structure of one. Using idioms can guide a speed-up BN development process and lead to better quality BNs [37].

To intuitively illustrate how the structure that we construct following idioms can be specified for further calculation, we consider actual probability values as an example. We have used Poisson distributions to simulate the number of introduced defects, assuming when the coding effectiveness is standard the parameter of the Poisson distribution would be ten, while when the coding effectiveness is advanced the parameter would be three. This represents a simplified assumption that the better coding effectiveness tend to introduce the less defects. The number of detected defects is simulated by Binomial distributions which is parameterized by the number of trials (n) and the probability of success (p). We use the number of introduced defects as n while p is set to be 0.9 when the testing effectiveness is advanced and is set to be 0.6 otherwise. The instantiated BN is shown in Figure 3-4.

Figure 3-4 An example of idioms in a BN



Experts' knowledge is widely used to specify the probabilistic weight between variables in a BN. For BNs which use CPTs to represent probabilistic weights, since the CPTs can be purely elicited from experts' knowledge, the concern about how the CPTs can precisely reflect the conditional relationship between variables has risen. Various of probability elicitation approaches, such as verbal and numerical probability scale [74] and frequency formats [75] have been proposed to tackle with this concern. Another concern is that, with the increasing number of states of the parent nodes, the states of child node will increase exponentially, which will become an exhausting and expensive process to elicit from experts. To simplify the elicitation process, [49] proposes Noisy-OR to encode expertise in complex CPTs. By assuming the effects of parent nodes are independent, Noisy-OR reduce the parameter number of the CPTs. A leak probability was often added in Noisy-OR as a dummy parent in the extended version of Noisy-OR. This dummy parent is always true in the model, representing all other causes that are not included in the model. Limitations is that it can only apply in model that only have Boolean variables. This restriction is tackled by the Noisy-Max proposed in [76].

When defining a causal structure is too complex for experts, structure learning can be applied to reveal the insights of underlying casual structure of data. Several algorithms about structure learning aiming to identify the dependencies between variables in BN using data have been proposed. They can be primarily classified into two categories:

Constraints-based algorithms: this kind of structure learning algorithms quantifies the dependencies between variables using conditional independence tests and accordingly removes or reorientates arcs between nodes to reconstruct a structure that satisfies these dependencies. Common conditional independence tests are introduced in [77] [78] [79].

Score-based algorithms: this kind of algorithms normally determines an optimal structure based on an objective function (also called scoring function) and a searching algorithm. A good objective function often rewards better goodness of fit and punish for additional arcs. Following by the searching algorithm, the optimal structure can be found when the the objective function is optimized. Related works include [80] [81] [82] [83].

3.3 Inference in Bayesian Networks

In a BN, when some variables have known states, the underlying inference algorithms can be used to update the probability distribution of the remaining variables, based on Bayes' theorem. Local exact inference can be executed in BNs which only have discrete variables or continuous variables that follows conditional Gaussian distributions [84]. However, it is impractical for models with mixture of discrete variables and continuous variables of non-standard distributions. Approximate inference has been studied to overcome such restrictions.

A list of approximate inference methods are discussed in [85]. One of the prevalent approaches is Gibbs sampling, which is a Markov chain Monte Carlo (MCMC) algorithm that approximates a specified multivariate probability distribution to obtain a sequence of observations when direct sampling is difficult. The work proposed in [86] extends this approach to include non-random variables computed

from other variables deterministically, which is implemented in software BUGS [87]. However, the main disadvantage of sampling algorithms is speed: they are often significantly slower than deterministic methods, making them unsuitable for large models and/or large data sets [85]. Moreover, the sampling method do not provide efficient mechanism for updating the model using newly observed information.

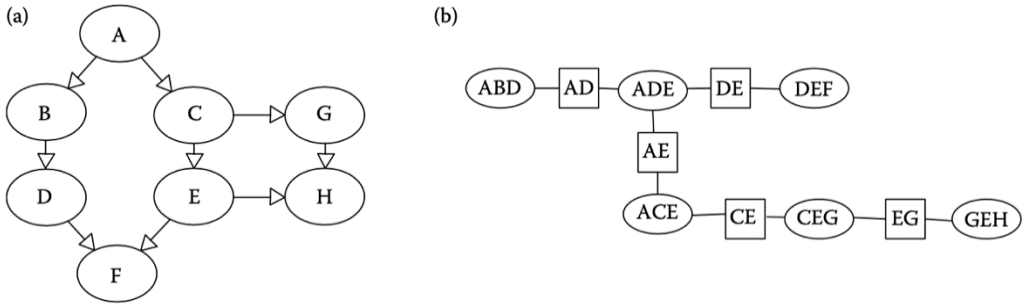
Another dominant approach is discretisation. Static discretisation allows approximate inference in an HBN. In this algorithm, the distribution of a continuous variable is discretized into pre-defined finite set of states for the further calculation. Inspired by the work on using non-uniform discretisation in a hybrid BN from [88], Dynamic Discretisation (DD) is proposed and used as an approximate inference algorithm in the work [50]. Continuous variables are dynamically discretised, with narrower intervals are assigned to the probability distribution where values are changing faster. By dynamically discretizing the distributions, more accuracy can be achieved in the regions that matter and less storage space is required compared with static discretization. Moreover, this kind of discretization can be adjusted anytime in response to new evidence to achieve greater accuracy.

The number of discretized states in variables can be large. So as in most real-world scenarios, where the number of variables and the states for each variables is large, calculations become daunting or impossible to do manually and computational complexity of inference in BN was found to be a “NP-hard” problem [37] [89].

Algorithms published in the late 1980s by researchers such as Lauritzen, Spiegelhalter, and Pearl has dramatically changed things. These algorithms provide efficient propagation for a large class of BN models [90] [48] [49]. They are efficient because they exploit the BN structure, by using a process of variable elimination, to carry out modular calculations rather than require calculations on the whole joint probability model. Among them, the most standard algorithm is called the junction tree algorithm. In this algorithm, a BN is firstly transformed into an associated tree structure, the junction tree, following a serial of procedures. We

borrow an example from [37] to show a BN and the correspond junction tree in Figure 3-5.

Figure 3-5 An example of (a) a BN and (b) the corresponding junction tree



The junction tree algorithm is designed to create the tree induced following the variable elimination process where the tree contains clusters (represented as round nodes and correspond to a set of the variables) and edges connecting the clusters, separated by separators (represented as square nodes and are formed by shared variables between clusters) as is shown in Figure 3-5 (b). Based on the constructed junction tree, we can carry out the local computations on parts of the tree. These calculations can then be propagated to efficiently get global answers and obtain queries asking for the marginal distributions from a BN. Details of constructing a junction tree from a BN and the *message passing* calculations which are performed using the algebra of CPTs for marginalization, division, and multiplication of tables are provided in [37].

The algorithms, dynamic discretisation and junction tree, have been implemented in the BN tool AgenaRisk [91]. The models built within this thesis were developed using AgenaRisk and its API for its flexibility of modelling and efficiency of calculation.

3.4 Influence Diagrams and Decision Trees

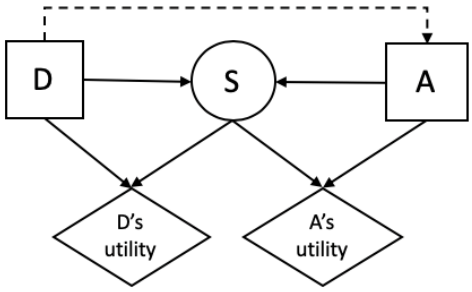
Decision problems can be analysed, structured and represented by an Influence Diagram (ID), which is a generalization of a Bayesian network [37]. In this section, we provide a general introduction of ID, and how to conduct decision analysis through ID using Decision Trees (DTs).

An ID is a special type of BN models that represents the interaction between decisions, chance variables, and utilities along with an algorithm to compute the expected utilities and identify those decisions that optimise the utility [37] [64]. In an ID, nodes represent each variable, with the convention that rectangles represent decisions, ovals represent chance variables, and diamonds represent utilities. Each decision node represents a decision, chance nodes represent random variables, which can be observable or non-observable, and utility nodes represent the pay-off for the decision maker. There is always an "ultimate" utility node that we seek to optimize. If multiple utility nodes are involved, they can be summarized into one ultimate node for the further calculation. An ID with chance, decision, and utility nodes that can involve mixtures of discrete and continuous variables are called Hybrid IDs (HIDs) which is a kind of HBNs.

Here we provide an example of ID in the cybersecurity context as a preliminary illustration of ID. More advanced ID for CRA is introduced and explored in Chapter 6 and Chapter 7.

There is an ID representing the interplay between an attacker (i.e., he) and a defender (i.e., she) towards some information asset as shown in Figure 3-6. This ID represents a sequential D-A game where the defender would make her defence deployment decision (represented by node D) first. This can then be observed by the attacker, and he would use this information to optimise his attack decision, i.e., whether to attack or use how much resource for attacking (represented by node A). 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 while A and S determine $A's$ utility. The decision that maximizes $D's$ utility would be the optimal strategy for the defender. The rule for the attacker is similar.

Figure 3-6 Influence diagram for the sequential Defend-Attack game



Generally, in an ID, incoming arcs to chance or utility nodes represent causal, deterministic, or associational relations between the node and its parents. Incoming arcs to decision nodes (shown by a dashed line) are "informational" arcs, representing the possibility that the temporal ordering, when the state of any parent node might be known before a decision is made. Informational (dashed line) arcs also specify the sequential order of decisions and observations. An ID cannot be computed without a strict sequential order.

After constructing an ID for a decision analysis problem, we can construct a Decision Tree (DT) [37] [64], which can be conducted using AgenaRisk, to represent all the potential decisions and their corresponding utility values. The decision which corresponds to the maximum (in general) utility would be determined to be the optima from the decision tree. We expand this in detail in Chapter 6.

3.5 Bayesian Networks for Cybersecurity Risk Analysis

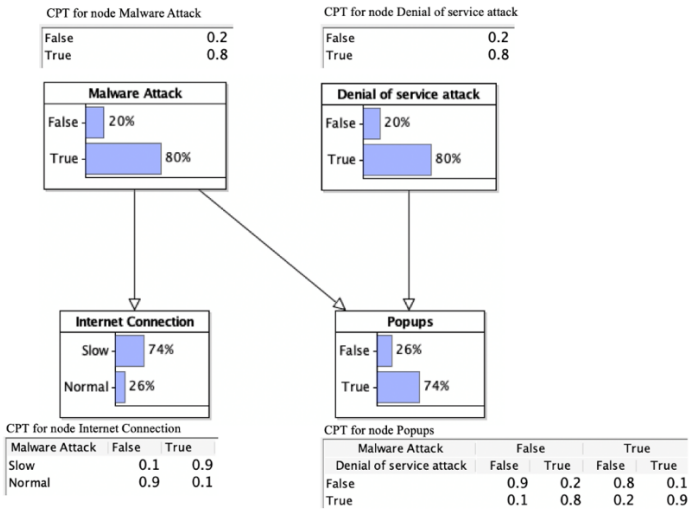
In this section, we use an example to provide a sense of how BNs can be adopted for modelling and calculating cybersecurity risks. Also, we interpret an existing CRA model, the kill chain, using a Bayesian network to illustrate how causal analysis can be adopted in the cybersecurity context.

3.5.1 A Toy Bayesian Network for CRA

We build a toy BN model to illustrate how BNs can be adopted to model personal belief of cyber events, as well as updating the belief while new evidence can be obtained. The context of this model is borrowed from [54].

In this case, we consider four events (variables) including two cyber-attacks which are “Denial of Service Attack” and “Malware Attack”, and two symptoms which are “Internet Connection” and “Pop-ups”. We represent the dependent relationships between these four events and the related CPTs in Figure 3-7. It is assumed that, towards a certain information system which is vulnerable, the probabilities of the system being attacked from “Denial of Service Attack” and “Malware Attack” are the same which are 0.8. The Internet connection can become slow given the system gets “Malware Attack”. We assume that the probability of the internet connection getting slower is 0.9, given “Malware Attack” is conducted. This is represented by the CPT of “Internet Connection”. Based on the same idea, the CPT of Internet “Pop-ups” can also be interpreted. The marginal distributions calculated in the BN, represent probabilities of “Internet Connection” being slow and “Pop-ups” based on the aforementioned belief setting.

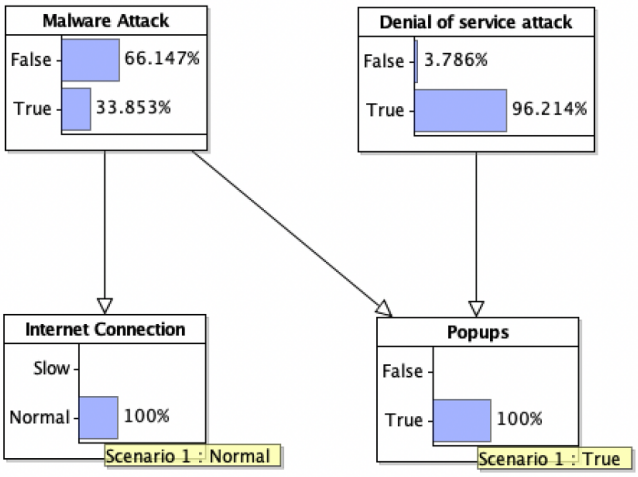
Figure 3-7 A toy cybersecurity risk BN model-the original model



The BN can also be used to determine the probabilities of the two kinds of attacks if certain symptoms appear and can be used to update the belief. Given symptoms, the BN can be used to compute the posterior probabilities of cyber-attacks. In this case, we assume that we observe that “Internet Connection” is “Normal”, and “Pop-ups” is “true”. By entering these new observations to the BN, the BN can compute the posterior probabilities of the other nodes “Denial of Service Attack” and “Malware Attack”. The BN model shown in Figure 3-8 represents that the presence

of pop-ups and normal internet connection is more likely caused by a Denial-of-Service attack rather than by a Malware attack.

Figure 3-8 A toy cybersecurity risk BN model-updated using new observations



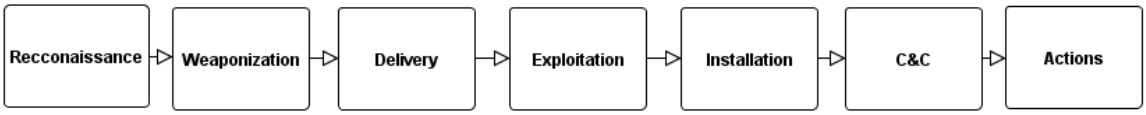
3.5.2 Intrusion Kill Chain and Bayesian Networks.

A kill chain is a systematic process which represents how an adversary can conduct actions to achieve desired effects. It is used by the U.S. military at first. The kill chain is expanded to a cyber kill chain model which is used especially on analysing intrusion of computer systems/network by Hutchins [42]. The model focuses on modelling actual attack steps and identifying optimal defence strategy on hypothetical attacks. In this section, we implement the kill chain using Bayesian networks to illustrate how causal analysis can be applied in cybersecurity risk modelling.

According to Hutchins, there are seven phases in the cyber kill chain as represented in Figure 3-9. Reconnaissance is the first phase which represents the attacker identifying and selecting the target. Weaponization means coupling a remote access Trojan and exploiting into a deliverable payload. Delivery means transmission of the weapon to the targeted environment. The three most prevalent delivery vectors for weaponized payloads are email attachments, websites, and USB removable media. After the weapon is delivered to the victim host, exploitation can trigger intruders' code. Most often, exploitation targets an application or operating system vulnerability, while it could also simply exploits the users themselves or leverage

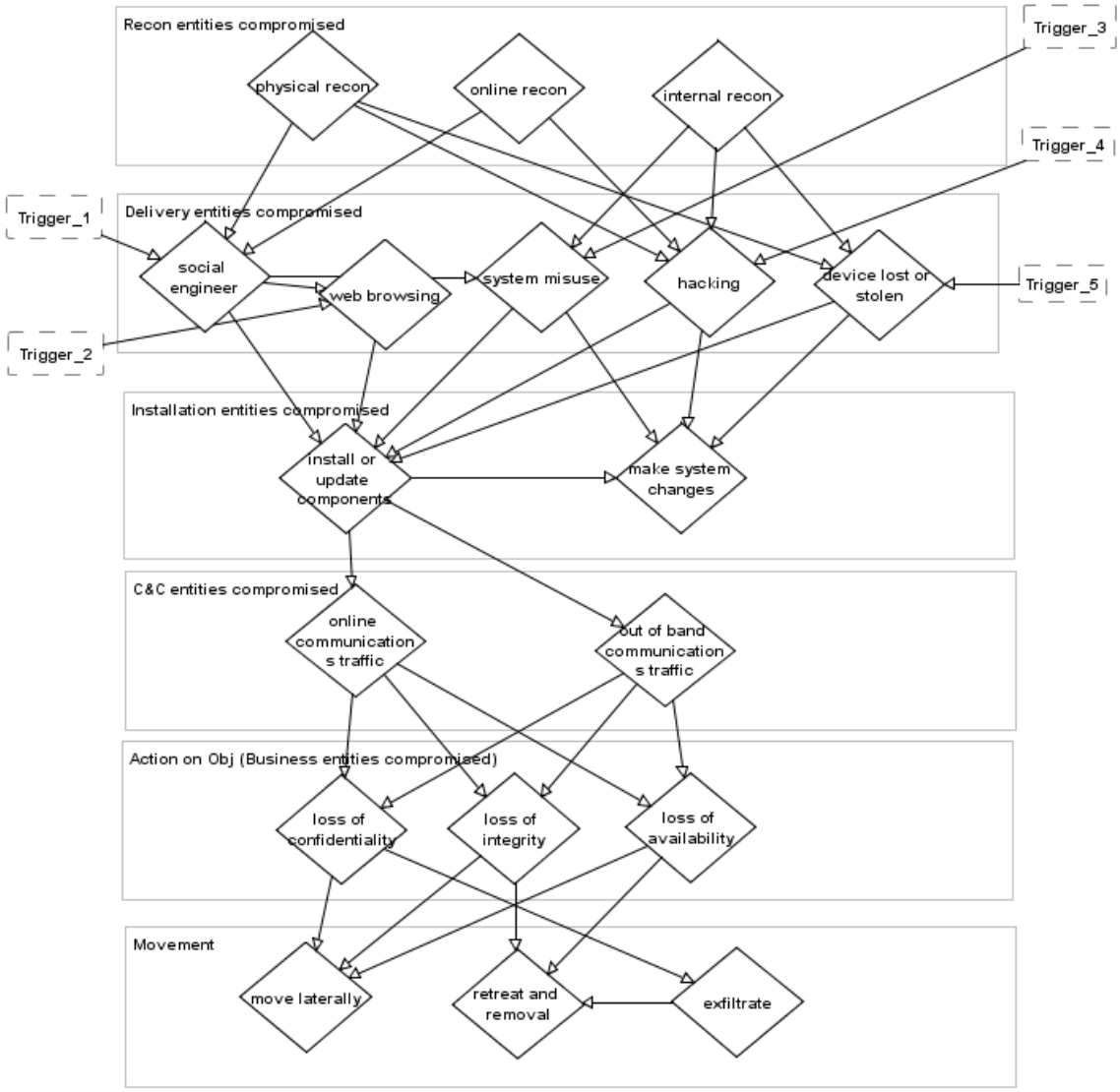
an operating system feature that auto-executes code. Installation of a remote access Trojan or backdoor on the victim system allows the adversary to maintain persistence inside the environment. Typically, compromised hosts must beacon outbound to an Internet controller server to establish a Command and Control (C&C) channel. Threat malware especially requires manual interaction rather than can take actions automatically. Once the C&C channel being established, intruders have "hands on the keyboard" inside the target environment. Only after progressing through the first six phases, can intruders take actions to achieve their ultimate objectives. Typically, this objective is data exfiltration which involves collecting, encrypting and extracting information from the victim environment.

Figure 3-9 Kill chain phases



In a kill chain, consequences are results of multiple interacting causes, which can be naturally represented by BNs. Compromising events which occur on physical or logical entities in certain kill chain phases are connected according to their chronological and causal relationship. We use an example borrowed from [37] to show how the kill chain can be transformed and instantiated using a BN in Figure 3-10. We use diamond nodes to represent compromised entities, arcs to represent causal precondition for compromised entities. We assume that only if the precondition occurs as well as there is a successful trigger the entity could be compromised. The process that attackers use their capability to overcome a vulnerability within an entity can be represented by trigger events. One or more trigger events can be connected with each entity. Trigger events of entities in delivery stage are displayed as examples while others are implicit. This is an instantiation and the extended version of the causal risk model illustrated by Figure 2-2.

Figure 3-10 A Bayesian Network Implementing Cyber Kill Chain

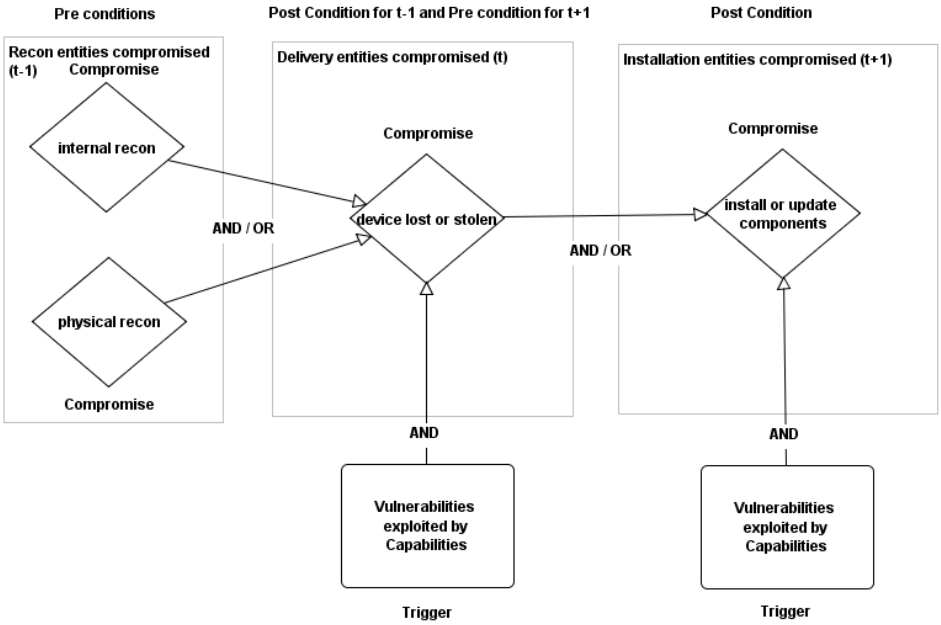


The state of an attack can be modelled by the transitions through the graph in the causal direction. Successful state transition from one comprised entity to another depends on that the previous entities in the chain are compromised and meanwhile at least one trigger events within the entity occurs. Here we use Boolean logic to define dependence between events and state transitions:

For preconditions and consequent events, we use AND / OR to represent state transitions; for trigger events, we always use AND, since without trigger no transition can occur. Here we simplify a part of the BN to demonstrate the state

transition with using AND gate to construct the state transition path as an example, which is shown in Figure 3-11.

Figure 3-11 State transition in the Bayesian Kill Chain



We set prior probabilities of the BN following Figure 3-12. Assuming we have obtained the evidence that internal recon has been compromised, then the posterior probabilities of delivery and installation entities being compromised are revised. The probability of "device lost or stolen" increases from 0.025% to 2.5% and the probability of "install or update" increases from 0.013% to 1.25%. The updated BN is shown in Figure 3-13:

Figure 3-12 Prior Probabilities in the BN

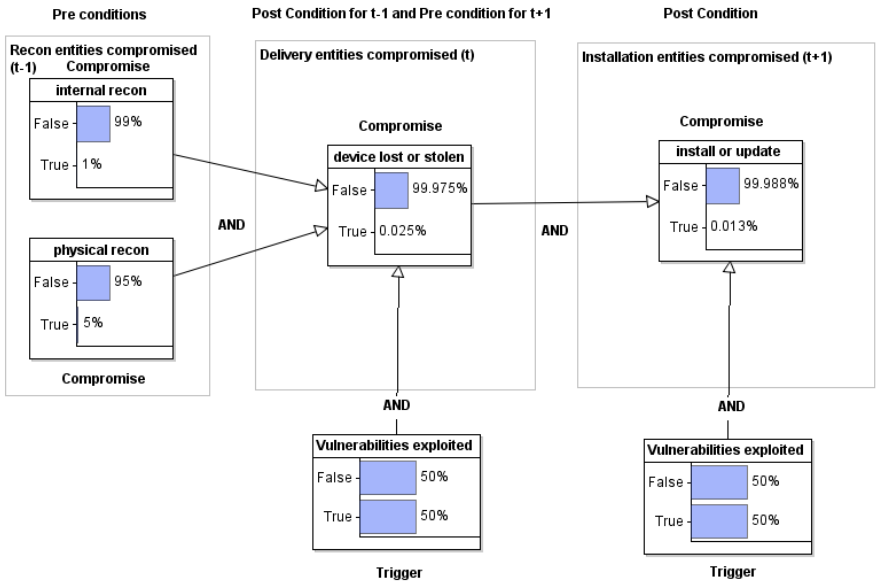
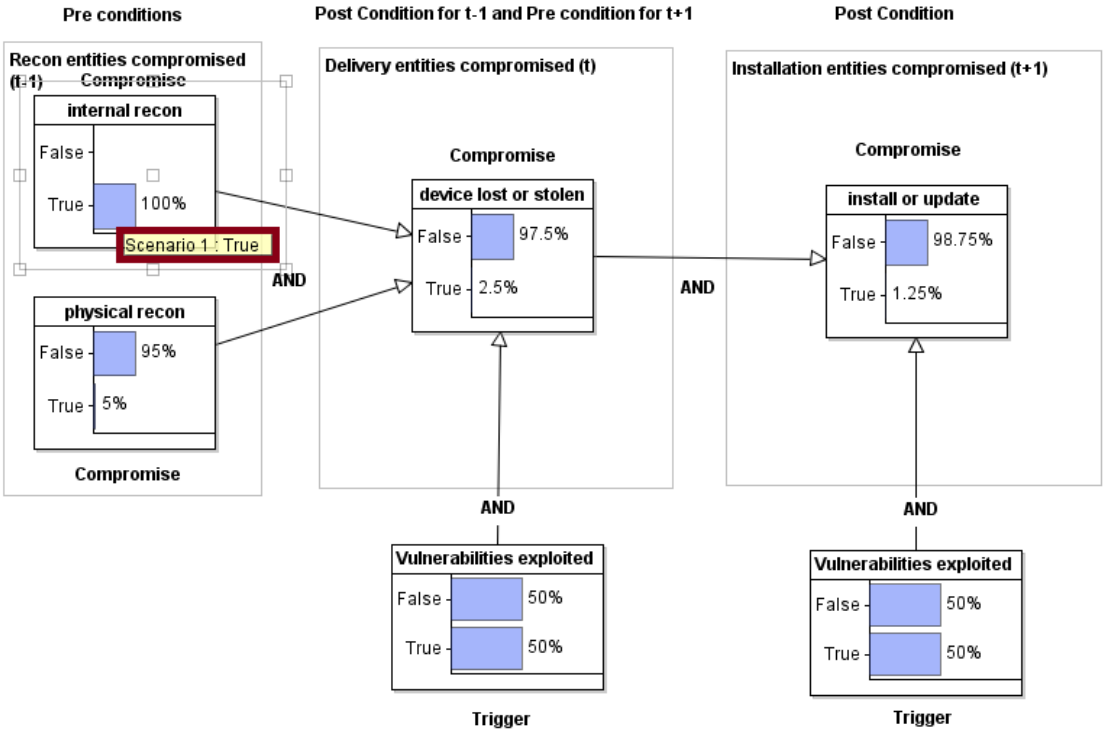


Figure 3-13 Revised probabilities in the BN.



Chapter 4 Analysis of the FAIR Model

This chapter provides an introduction of Factor Analysis of Information Risk (FAIR). An overview is provided in section 4.1 for constructing a preliminary understanding of FAIR and its position in CRM and revealing a number of important limitations that FAIR has. Section 4.2 and 4.3 provide a detailed analysis of the assumptions of the FAIR model, focusing on its taxonomic structure and algorithms while Section 4.4 and 4.5 provide an in-depth analysis of the calculation mechanism of the FAIR model, which has hitherto not appeared in the literature. The analysis provided in this chapter constructs the motivation and foundation of constructing FAIR-BNs in Chapter 5.

4.1 Introduction of FAIR and Our Related Work

Factor Analysis of Information Risk (FAIR) is a well-known CRA framework [28, 29] and has been widely applied and recognized in academic research [92] [93] [94] [95] and industry [32] [96]. We adopt FAIR as a foundation and the benchmark to conduct organizational risk analysis in our research. Another attempt improving the flexibility of FAIR by using BNs was proposed in [93], which is consistent with 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 propose a complete implementation of the FAIR model with BNs.

A framework for estimating the completeness of information security risk assessment methods is constructed in [10]. This work investigates CRA methods which cover the CRA methods that we have reviewed in section 2.2 with finding that the “ISO/IEC 27005 Information Security Risk Management” tackles with the most comprehensive aspects for CRA and FAIR tackles with the most comprehensive aspects in cybersecurity risk estimation. In terms of ISO/IEC 27005, it specifies more details in risk management based on ISO/IEC 27001 which outlines the process for managing risk at a fairly high level, although without providing specifics or identifying a methodology for calculating risks. The FAIR

methodology can be used in the context of ISO/IEC 27005 standard and can be a supplement to ISO/IEC 27005 and other CRA standard/framework for enhancing the risk estimation [97].

More precisely, FAIR provides the means to determine and articulate risks and furthermore enhancing the risk estimation following four stages: 1) identification of (information) assets and threats to them; 2) evaluation of loss event frequency based on identification of its sub-factors; 3) evaluation of probable loss magnitude based on identification of its sub-factors; 4) derivation of risks which represented by financial losses. Figure 4-1 illustrates how FAIR supplements the CRA process proposed in ISO/IEC 27005 [6].

Figure 4-1 FAIR's place in ISO/IEC 27005 [97]

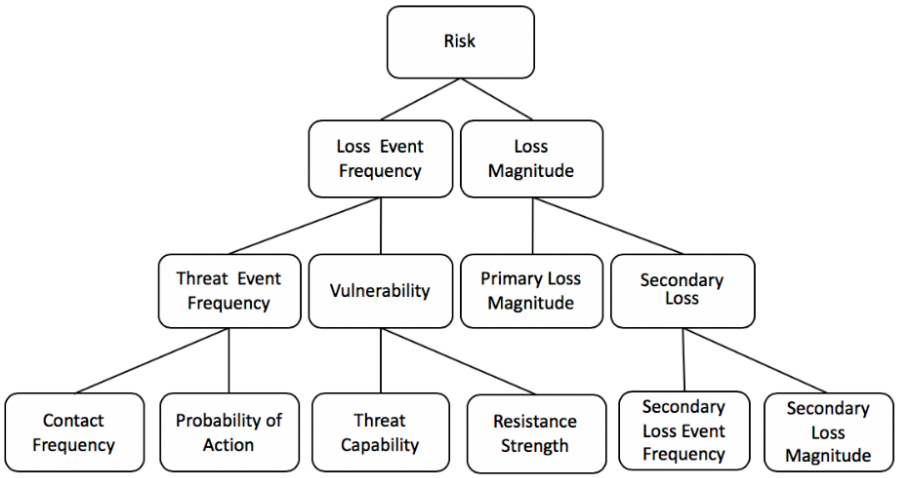
7.0	Context Establishment	
7.1	General Considerations	
7.2	Basic Criteria	
7.3	Scope and Boundaries	
7.4	Organization of Information Security Risk Management	
8.0	Information Security Risk Assessment	
8.1	General Description of Information Security Risk Assessment	
8.2	Risk Analysis	**Risk Analysis using FAIR**
8.2.1	Risk Identification	Stage 1:
8.2.1.1	Introduction to risk identification	Identify scenario components
8.2.1.2	Identification of assets	Identify the asset at risk
8.2.1.3	Identification of threats	Identify the threat community
8.2.1.4	Identification of existing controls	Stage 2:
8.2.1.5	Identification of vulnerabilities	Evaluate Loss Event Frequency (LEF)
8.2.1.6	Identification of consequences	Estimate probable Threat Event Frequency (TEF)
8.2.2	Risk estimation	Estimate Threat Capability (TCap)
8.2.2.1	Risk estimation methodologies	Estimate Control Strength (CS)
8.2.2.2	Assessment of consequences	Derive Vulnerability (Vuln)
8.2.2.3	Assessment of incident likelihood	Derive Loss Event Frequency (LEF)
8.2.2.4	Level of risk estimation	Stage 3:
8.3	Risk Evaluation	Evaluate Probable Loss Magnitude (PLM)
		Estimate worst-case loss
		Estimate Probable Loss Magnitude (PLM)
		Stage 4:
		Derive and articulate risk
9.0	Information Security Risk Treatment	
9.1	General Description of Risk Treatment	
9.2	Risk Reduction	
9.3	Risk Retention	
9.4	Risk Avoidance	
9.5	Risk Transfer	
10.0	Information Security Risk Acceptance	
11.0	Information Security Risk Communication	
12.0	Information Security Risk Monitoring and Review	
12.1	Monitoring and Review of Risk Factors	
12.2	Risk Management Monitoring, Reviewing, and Improving	

Based on the definition that risk is the "the probable frequency and magnitude of future loss", FAIR provides a taxonomy framework which breaks down cybersecurity risk into the frequency and loss magnitude of cyber events, and then further sub-factors. Moreover, it builds look-up tables for qualitatively reasoning severity of risks [28]. Based on FAIR, the OPEN Group establishes a quantitative

FAIR model based on the FAIR taxonomy structure [98, 99]. The FAIR model which integrates a set of quantitative reasoning algorithms into the FAIR structure identifies how risk factors interact with each other and ultimately predicts financial losses of cyber events [98].

More precisely, to structure risk analysis, FAIR uses a taxonomy to classify risk (financial loss) into the frequency and loss magnitude of cyber events, and then further sub-factors and represent the relationships between these risk factors as shown in Figure 4-2. More detailed introduction of this is provided in section 4.2.

Figure 4-2 Taxonomy structure of the FAIR model



FAIR covers more aspects of CRA compared to other prominent CRA frameworks [10]. 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 [100] [101].

Based on analysis and experiments that we've done in the following sections and Chapter 5, we unpick the assumptions and algorithms used in the FAIR model and identify a number of potential serious limitations.

A triangular distribution is a probabilistic distribution with lower limit a, upper limit b and mode c, where $a < b$ and $a \leq c \leq b$ [37]. One limitation of the FAIR

model is that the FAIR model can only use triangular distributions to simulate input risk factors (which are variables without parent variables, i.e., the primary loss magnitude) of the model and use fixed functions to determine relationship between input variables and their child variables (we summarize these functions in table 4-1). This is because of the tailored algorithms that FAIR model has adopted to produce high-efficient calculation. More precisely, the FAIR model simplifies the simulation-based calculation using cached data combined with a statistical approximation technique which involves constructing a kind of quantile distribution function, Bounded Metalog Distribution (BMD). The BMD is constructed based on triangular distributions and is then applied for simulating the ultimate risk (the total loss face by the organization) in the FAIR model. We introduce how the BMD is constructed and applied in the FAIR model in section 4.4 and 4.5.

The issue with triangular distribution, that only triangular distributions can be used as inputs for the FAIR model, is related to implementation (not theory). This is because only when input variables follow triangular distribution, the tailored algorithms for high-efficient calculation are valid. We have also implemented the FAIR's assumptions (about risk factors and their dependent relationship) using FAIR-MC, which conducts simulation-based calculation straightforward without employing the BMD approximation nor using cached data. The FAIR-MC does not have the limitation of input distributions as the FAIR model.

Since only triangular distributions can be used for modelling input variables in the FAIR model, input factors which might be depicted more precisely by alternative statistical distributions for certain cases may be poorly approximated. For example, long-tailed distributions [102] [103] would be more suitable for modelling the frequency of Advanced Persistent Threat (APT) [104] than using triangular distributions. The restriction of the FAIR model, that using triangular distributions for modelling inputs (the frequency of APT in this case), could therefore introduce inaccuracy. We provide detailed experimental analysis for this in section 5.3.3. Moreover, the FAIR model is difficult to extend 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 HBNs. Furthermore, by employing BNs, we can connect the FAIR model with other advanced CRA models to enhance the original model, for example to analyse interaction 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 [45], and from the game-theoretic perspective, such as Adversarial Risk Analysis (ARA) [105]. In Chapter 8, we have provided examples to illustrate how the other CRA models can be incorporated the FAIR-BN to provide the integrated risk assessment and management solution. We call them Extended FAIR-BNs (EFBNs). More details of the taxonomic structure and the calculation mechanism of the FAIR model are introduced in the following sections.

4.2 FAIR Model Structure: Taxonomy and Aggregation

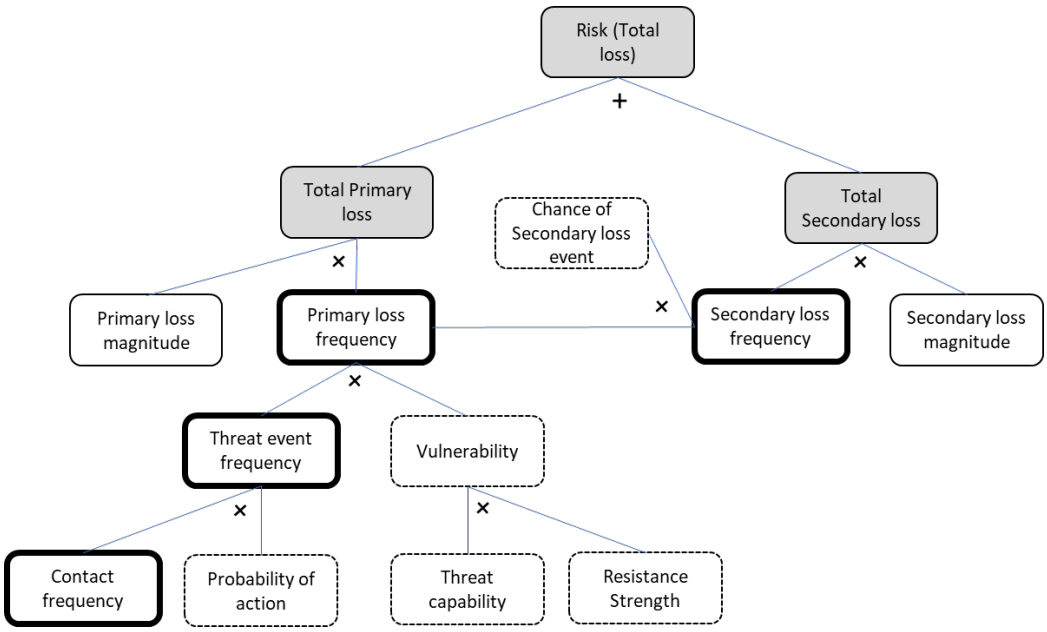
The taxonomy structure of the FAIR model [29, 100] was shown in Figure 4-2, with the risk classes being modelled. Risk (financial loss) is defined by Loss Event Frequency (LEF) and Loss Magnitude (LM). LEF is defined as the frequency that a threat agent will inflict harm on an information asset within 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’ [28]. 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 [100]). 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).

Figure 4-3 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 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.

Figure 4-3 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)



4.3 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 [100]. We summarize the factors and functions in Table 4-1.

Table 4-1 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	$L_T = L_P + L_S;$
		Secondary Total Loss (STL): L_S	
2	Primary/Secondary Loss (PL/SL): L_P/L_S	Loss Event Frequency (PLEF/SLEF) : F_P/F_S	$L_P = RA(F_P, LM_P);$ $L_S = RA(F_S, LM_S);$
		Loss Magnitude (PLM/SLM): LM_P/LM_S	
3	Mean of Primary Loss Event Frequency (MPLEF): M_{PLEF}	Threat Event Frequency (TEF): F_{TE}	$M_{PLEF} = F_{TE} \times P_V;$
		Vulnerability (V): P_V	
4	Primary Loss Event Frequency (PLEF): F_P	Mean of PLEF (MPLEF): M_{PLEF}	F_P = Poisson ($\lambda = M_{PLEF}$);
5	Secondary Loss Event Frequency (SLEF): F_S	Primary Loss Event Frequency (PLEF): F_P	$F_S = Binomial\ (n = F_P, p = P_{SL});$
		Chance of Secondary Loss (CSL): P_{SL}	
6	Threat Event Frequency (TEF): F_{TE}	Contact Frequency (CF): F_C	$F_{TE} = F_C \times P_A;$
		Probability of Action (PoA): P_A	
7	Vulnerability (V): P_V	Threat Capability (Tcap): P_{TC}	$P_V = P(P_{TC} > P_{RS}).$
		Resistance Strength (RS): P_{RS}	

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 [101]. 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 4-1 can be performed on corresponding sample vectors.

Risk assessment through the FAIR model includes two procedures: assessing loss event frequencies (calculating factors 3-7 referred to Table 4-1) and aggregating loss magnitudes using assessed frequencies to calculate the total loss (calculating factors 1-2 referred to Table 4-1). 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, [67] [106]. 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 [107], 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 statistical 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 approximating L_P. This kind of adopted quantile distribution is named Bounded Metalog Distribution (BMD) [108]. 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 Section 4.4 and demonstrate how the FAIR model uses BMDs and cached data to produce risk aggregation results in Section 4.5.

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 chains or that derived from adversarial risk analysis. We propose using Bayesian Networks (BNs) as an alternative way to implement and eliminate restrictions of the FAIR model in Chapter 5 and extend the FAIR model with other CRA models in Chapter 8 to show the expandability of the FAIR-BN and its potential to provide an integrated CRA solution.

4.4 The Bounded Metalog Distribution

In this section, we provide details of how the Bounded Metalog Distribution (BMD) is constructed within the FAIR model. A 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) [108], 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 [108]: The Metalog distribution of a random variable *M* with *n* terms is:

$$M_n(y;\boldsymbol{x},\boldsymbol{y}) = \tag{4-1}$$

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

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 $\mathbf{a} = (a_1, \dots, a_n)$is given by

$$\mathbf{a} = [\mathbf{Y}_n^T \mathbf{Y}_n]^{-1} \mathbf{Y}_n^T \mathbf{x} \tag{4-2}$$

Where $\mathbf{Y}_n^T$ is the transpose of $\mathbf{Y}_n$, whilst the $m \times n$ matrix $\mathbf{Y}_n$ is:

$$\mathbf{Y}_n = \tag{4-3}$$

$$\begin{bmatrix} 1 & ln\left(\frac{y_1}{1-y_1}\right) \\ & \vdots \\ 1 & ln\left(\frac{y_m}{1-y_m}\right) \end{bmatrix} \quad \text{for } n = 2$$

$$\begin{bmatrix} 1 & ln\left(\frac{y_1}{1-y_1}\right) & (y_1 - 0.5)ln\left(\frac{y_1}{1-y_1}\right) \\ & & \vdots \\ 1 & ln\left(\frac{y_m}{1-y_m}\right) & (y_m - 0.5)ln\left(\frac{y_m}{1-y_m}\right) \end{bmatrix} \quad \text{for } n = 3$$

$$\begin{bmatrix} 1 & ln\left(\frac{y_1}{1-y_1}\right) & (y_1 - 0.5)ln\left(\frac{y_1}{1-y_1}\right) & (y_1 - 0.5) \\ & & \vdots & \\ 1 & ln\left(\frac{y_m}{1-y_m}\right) & (y_m - 0.5)ln\left(\frac{y_m}{1-y_m}\right) & (y_m - 0.5) \end{bmatrix} \quad \text{for } n = 4$$

$$\begin{bmatrix} \mathbf{Y}_{n-1} & (y_1 - 0.5)^{\frac{n-1}{2}} \\ & \vdots \\ & (y_m - 0.5)^{\frac{n-1}{2}} \end{bmatrix} \quad \text{for odd } n \geq 5$$

$$\begin{bmatrix} \mathbf{Y}_{n-1} & (y_1 - 0.5)^{\frac{n}{2}-1} ln\left(\frac{y_1}{1-y_1}\right) \\ & \vdots \\ & (y_m - 0.5)^{\frac{n}{2}-1} ln\left(\frac{y_m}{1-y_m}\right) \end{bmatrix} \quad \text{for even } n \geq 6$$

The proof of that the quantile function of a random variable M can be parameterized by points on the CDF of M is provided in [108].

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

Definition 2 [108]: 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 $z = ln\left(\frac{x-b_l}{b_u-x}\right)$ is Metalog-distributed.

Setting $ln\left(\frac{x-b_l}{b_u-x}\right)$ equal to (4-1) and solving for x yields the BMD function with n terms:

$$M_n^{logit}(y; \boldsymbol{x}, \boldsymbol{y}, b_l, b_u) = \begin{cases} \frac{b_l + b_u e^{M_n(y)}}{1 + e^{M_n(y)}} & 0 < \mathrm{y} < 1 \\ b_l & \mathrm{y} = 0 \\ b_u & \mathrm{y} = 1 \end{cases} \tag{4-4}$$

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 (4-4), 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 the following section.

4.5 Application of BMD in Risk Aggregation

This section introduces how the FAIR model uses BMDs and cached data to produce risk aggregation results. 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 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. And 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 $r = \frac{M_{mid}-M_{min}}{M_{max}-M_{min}}$, and belongs to a predetermined ratio set, S = {0, 0.1, 0.2, ..., 0.9, 1, 1.01}. These L_P samples are firstly taken to average over the corresponding $\hat{f}_j$, 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, the result is a 27×12 data matrix, of which each element is a vector, $\boldsymbol{v}$, corresponding to a pair of $(\hat{f}_j,\ \hat{s}_k)$. This data matrix is prepared and provided by the FAIR model [99].

Based on the cached data, the FAIR model approximates the quantile value vector, $\boldsymbol{v}$, for the actual frequency sample f_i and a LM distribution of ratio, r, by interpolation based on stored 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 (4-5):

$$\boldsymbol{v} = \left(In\left(\frac{\boldsymbol{v_1}}{\mathbf{1}-\boldsymbol{v_1}}\right) \times a + In\left(\frac{\boldsymbol{v_2}}{1-\boldsymbol{v_2}}\right) \times (1-a)\right) \times b + \left(In\left(\frac{\boldsymbol{v_3}}{1-\boldsymbol{v_3}}\right) \times a + In\left(\frac{\boldsymbol{v_3}}{1-\boldsymbol{v_3}}\right) \times (1-a)\right) \times (1-b) \quad (4\text{-}5)$$

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

In formula (4-5), $\boldsymbol{v}$ is the quantile value vector which stores approximated quantile values corresponding to f_i and r, while $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$ 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 $\frac{M_{mid}-M_{min}}{M_{max}-M_{min}}$, 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 ($\boldsymbol{v}_1$, $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$) following formula (4-5). The approximated $\boldsymbol{v}$ is then used to specify the Metalog distribution [108] of $\bar{L}_P$. The Metalog distribution is a kind of logistic quantile distribution that can be determined by quantile values. For example, $\boldsymbol{v}$, 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 (4-6), which is referred to as Bounded Metalog Distribution (BMD) in [108]. We have described details of Metalog distribution and BMD in Section 4.4.

$$L_P(i) = f_i * (M_{min} + M_{max} \frac{e^{M_9(y)}}{1 + e^{M_9(y)}}) \quad (4\text{-}6)$$

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 (4-6). 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 $L_T = L_P + L_S$.

Chapter 5 Constructing and Evaluating the FAIR-BN

In this chapter, we construct an alternative of the FAIR model using Bayesian networks, called FAIR-BNs that incorporate the same modelling assumptions used by the FAIR model but also supports wider assumptions and can be more easily extended. This is introduced in Section 5.1. To perform accuracy evaluation for the FAIR model and FAIR-BNs, we construct a Monte Carlo (MC) simulation (FAIR-MC) without using the approximation techniques that applied by FAIR and introduce *J* divergence [109] [110] as the criteria in Section 5.2. In Section 5.3, the performance of the FAIR model and the proposed FAIR-BN is evaluated, meanwhile, cases in which the FAIR model produces inaccurate results are identified. Section 5.4 discusses pros and cons of the FAIR model, FAIR-BN and FAIR-MC.

5.1 The FAIR-BN

To address limitations of the original FAIR model, we develop a more flexible alternative approach, which we call FAIR-BN, to faithfully implement the FAIR model using HBNs. 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) can be easily extended by incorporating with other CRA models built using BNs to enhance the analysis (the second point is illustrated in Chapter 8).

We introduce how we implement FAIR using HBNs with focusing on revealing the mechanism of the RA process in subsection 5.1.1. The algorithm that we propose to implement RA using BNs based on the work [67] is introduced in subsection 5.1.2. An example is provided in subsection 5.1.3 to illustrate that the proposed FAIR-BN allows a wider range of distributions to represent and process input variables, compared with the original FAIR.

5.1.1 Constructing FAIR using HBNs

As explained in Section 4.3, 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. The basic idea of conducting RA is identical in the FAIR model and the FAIR-BN. In the implementation level, since the FAIR model adopted simulation-based calculation, the algorithm used for calculating L_p and L_s are the same, as is illustrated by the second entry of the Table 4-1. In comparison, the algorithms for calculating L_p and L_s are different in FAIR-BNs, since the probabilistic-inference-based calculation has been adopted. There are two types of risk aggregation (denoted as RA_1 and RA_2) needed in FAIR-BN as shown in Table 5-1.

Table 5-1 Types of risk aggregation process in FAIR-BN

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: $F_{P\&S}$	$L_T = RA_2\ (F_{P\&S}, LM_P, LM_S)$.

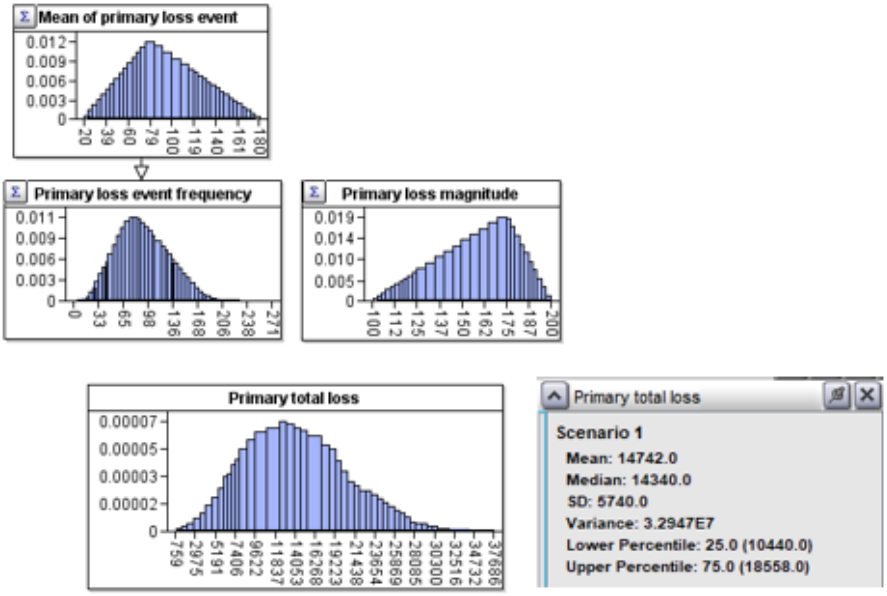
To show how RA_1 is implemented using BNs, we introduce the calculation of $L_p = RA_1(F_p, LM_p)$ as an example. This calculation is conducted using n-fold convolution [67] [106]. Assuming that, in a given period, a cyber event can happen *n* times where *n* is any number between 0 and the upper bound *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 (5-1):

$$L_P = P(0)LP_0 + P(1)LP_1 + P(2)LP_2 + \cdots + P(N)LP_N \tag{5-1}$$

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 Figure 5-1, we

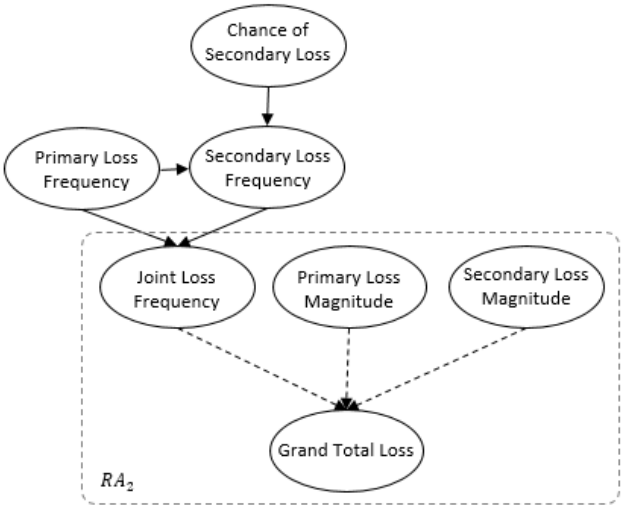
show a RA_1 result given input distributions for primary loss frequency and magnitude.

Figure 5-1 RA_1 result of FAIR-BN
with M_{PLEF} following $Triangular(min = 20, ml = 80, max = 180)$
whilst LM_P following $Triangular(min = 100, ml = 175, max = 200)$



The BN shown in Figure 5-2 models the relationships among associated variables involved in the risk aggregation process RA_2.

Figure 5-2 Risk factors involved in RA_2



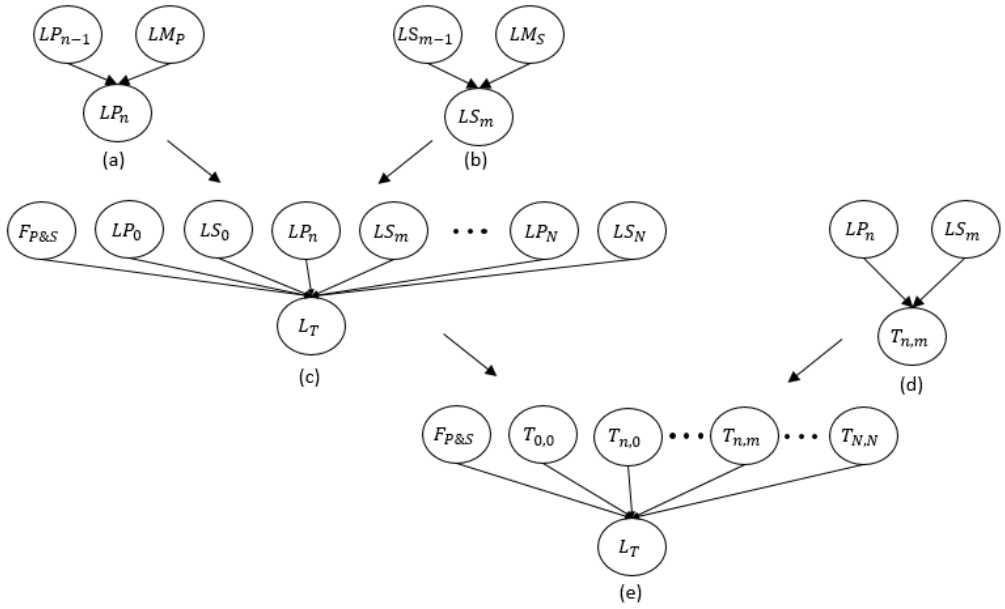
In the RA_2 process, the distribution of Total Loss (TL), L_T, can be calculated by conducting risk aggregation on the joint frequency $F_{P\&S}$ and the corresponding

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 (5-2):

$$L_T = \sum_{n=0}^{N}[\sum_{m=0}^{n} P(F_P = n, F_S = m) \times (LP_n + LS_m)] \qquad (5\text{-}2)$$

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 (a), (b) and (c) in Figure 5-3 to illustrate the RA_2 process represented by formula (5-2).

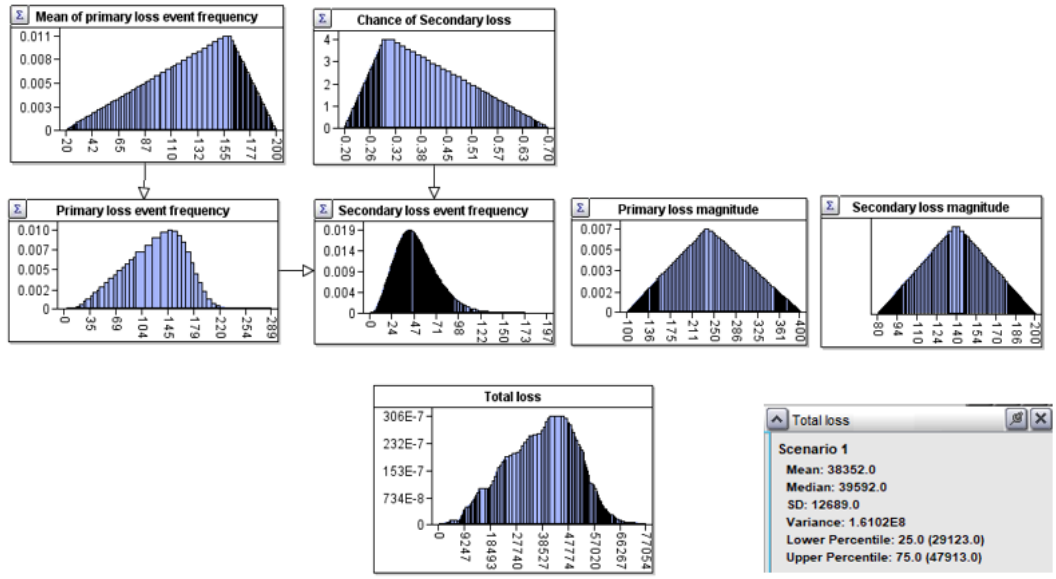
Figure 5-3 BNs used to implement the RA_2 risk aggregation process



We firstly simplify the BN (c) to BN (e) in Figure 5-3, 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 [67]. We have extended this 1-Dimension CDF method to a 2-Dimensions CDF method to implement risk

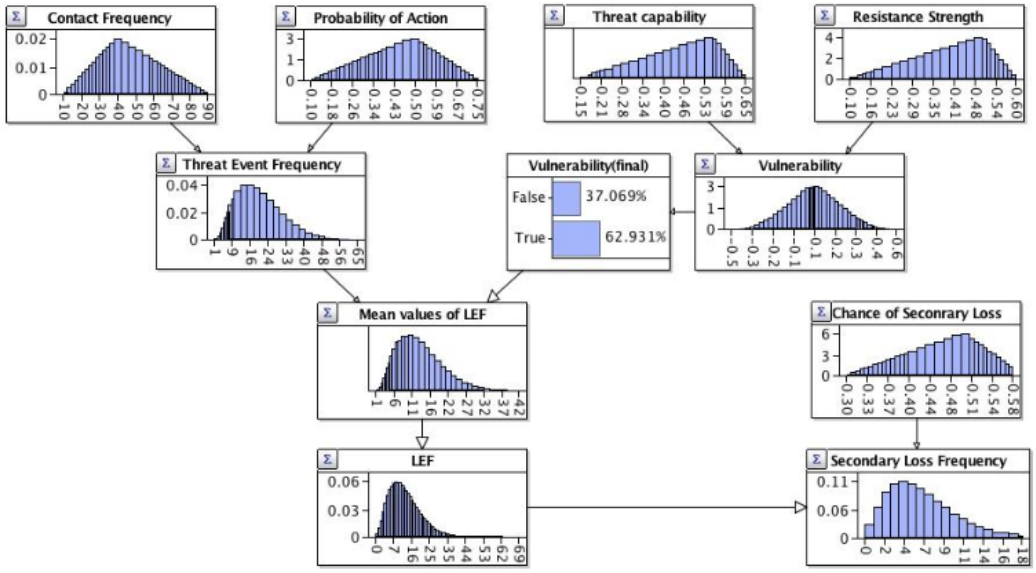
aggregation on the joint frequency distribution as the RA_2 process. We use AgenaRisk to implement the related algorithms which are described in subsection 5.1.2. An example result showing how RA_2 is calculated is shown in Figure 5-4.

Figure 5-4 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 $Triangular(min = 80, ml = 140, max = 200)$



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 Figure 5-5. 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.

Figure 5-5 FAIR-BN for calculating F_P and F_S



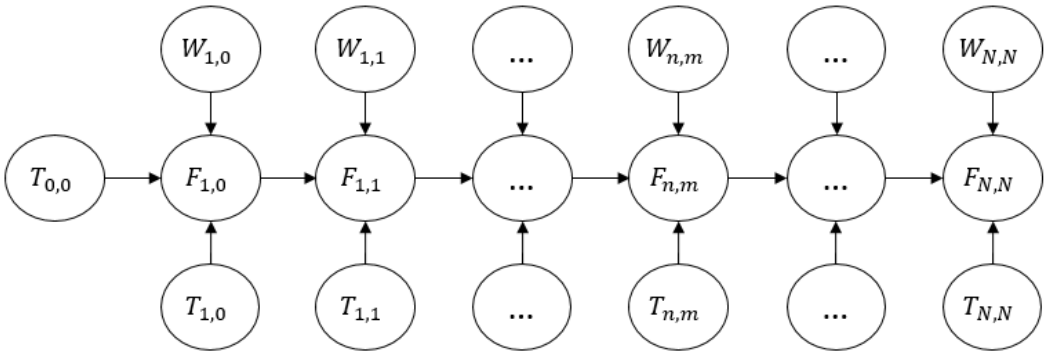
5.1.2 Factorization of the BN for $\boldsymbol{RA_2}$ Process

We have used Figure 5-3 to illustrate the RA_2 process represented by formula (5-2). The related calculation can be very space inefficient. We have addressed this by proposing an adjusted Compound Density Factorization (CDF) method for calculating RA_2 space efficiently based on the CDF method proposed to calculate RA_1 in [67]. More precisely, we have extended the 1-Dimension CDF method proposed in [67] to a 2-Dimensions CDF method to implement risk aggregation on the joint frequency distribution, which is the RA_2 process. We have implemented this adjusted CDF method by programming via Java API of AgenaRisk. In this section, we introduce the main idea of the adjusted CDF.

We demonstrate the adjusted CDF method in Figure 5-6. As is illustrated by Figure 5-3 (e), we have used $T_{n,m}$ to represent the total loss of which the primary loss event frequency is n (for $n = 1\ to\ N$) and the secondary loss event frequency is m (for $m = 1\ to\ n$). 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) in Figure 5-3 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. $\{T_{0,0}, T_{1,0}\}$,

$\{F_{1,0}, T_{1,1}\}$, …, $\{F_{n,m-1}, T_{n,m}\}$.,…, $\{F_{N,N-1}, T_{N,N}\}$. Factor variables, $F_{n,m}$, are created to calculate the weighted aggregation for each step.

Figure 5-6 Factorization of the BN for RA_2 process



The Conditional Probability Table (CPT) for $W_{n,m}$ is defined by the following:

$$P(W_{n,m} = true) = \frac{\sum_{i=0}^{n-1}\sum_{j=0}^{i} P_{i,j} + \sum_{j=0}^{m-1} P_{n,j}}{\sum_{i=0}^{n-1}\sum_{j=0}^{i} P_{i,j} + \sum_{j=0}^{m} P_{n,j}} \tag{5-3}$$

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

$$F_{n,m} = \begin{cases} F_{n,m-1} \text{ if } W_{n,m} = True \\ T_{n,m} \text{ if } W_{n,m} = False \end{cases} \tag{5-4}$$

Since $T_{0,0}$ and $T_{1,0}$ are mutually exclusive, the marginal distribution for variable $F_{1,0}$ is:

$$F_{1,0} = P(W_{1,0} = True)\, T_{0,0} + P(W_{1,0} = False)\, T_{1,0} \tag{5-5}$$

Similarly, the marginal for variable $F_{n,m}$ become:

$$F_{n,m} = P(W_{n,m} = True)\, F_{n,m-1} + P(W_{n,m} = False)\, T_{n,m} \tag{5-6}$$

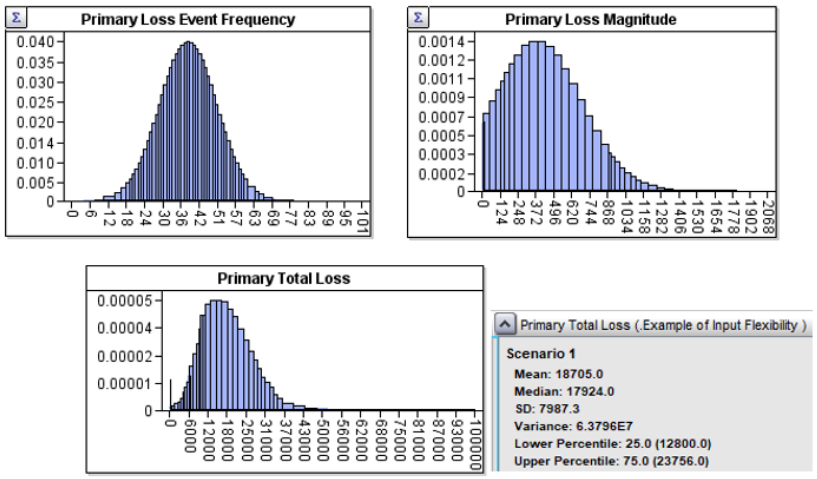
After factorizing the density aggregation process, we can calculate the marginal distribution of $F_{n,m}$ more efficiently following formula (5-6), which yields the risk aggregation result given primary and secondary loss frequencies and their loss magnitudes. We have implemented this method using AgenaRisk packages.

5.1.3 Flexibility of the FAIR-BN

The aim of this section is to show the flexibility of the FAIR-BN intuitively. In Section 3, we have noted that the FAIR model has significant restrictions in that only triangular distributions can be used as inputs by the end user and the model uses fixed statistical methods to deal with input data, despite the fact that the risk factors may be better represented by other, perhaps more diverse distributions in practice. The use of BNs eliminates the restrictions on the input factors, as well as on dealing with input data. 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).

Figure 5-7 shows a FAIR-BN model result achieved by computing RA_1 with Truncated Normal (TNormal) distributions [111] being assigned to primary loss event frequency, F_P, and the corresponding loss magnitude, LM_P, rather than triangular distributions 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.

Figure 5-7 A RA_1 result of FAIR-BN with PLEF and PLM following TNormal distributions
F_P follows $TNormal(\mu = 40, \sigma^2 = 100, LowerBound = 0, UpperBound = 200)$ whilst
LM_P follows $TNormal(\mu = 400, \sigma^2 = 100{,}000, LowerBound = 0, UpperBound = 10{,}000)$



In this section, we have intuitively illustrated that FAIR-BN is more flexible than the FAIR model, since the FAIR-BN can adopt a wider range of distributions to model input variables. In section 5.3.3, we have specified three practical cases in which distributions other than triangular distributions would be more suitable for modelling input variables. In these cases, FAIR-BNs have provided accurate calculation results while the FAIR model which can only use triangular distributions to model input variables shows less accuracy. The related experiments are provided in section 5.3.3 and are summarized in section 5.3.4.

5.2 Simulation and Evaluation Using Monte Carlo

We evaluate the quantitative accuracy of FAIR and FAIR-BN using results generated by the proposed FAIR-MC and the measurement *J* divergence [109] [110]. 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) [108] 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, meanwhile we use one thousand samples generated by FAIR-BN and the FAIR model respectively to represent the results from the two models. 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. Therefore, we used results from the FAIR-MC as the "gold standard" to evaluate the accuracy of the FAIR model and the FAIR-BN. In this thesis, *J* divergence [110] has been adopted as the measurement, of which the value reflects the "distance" between the results generated from the tested models and the "gold standard" generated from

FAIR-MC. Therefore, we can assume that the smaller *J* divergence the model has against the FAIR-MC, the more accurate the model is.

measure the distance between this standard and results (represented one thousand samples respectively) generated by the FAIR model and FAIR-BN using *J* divergence.

5.2.1 Implementing the FAIR Model by Monte Carlo

Monte Carlo (MC) methods are a broad class of computational algorithms that generate numerical results from repeated random sampling [112]. In this section, we describe how we use MC simulation methods to implement functions in the FAIR model as listed in Table 4-1 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-MC 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 (5-7):

$$L_P^i = \sum_{k=0}^{f_i} LM_P^k \tag{5-7}$$

where $i = 0, \dots, n$ 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 P_{SL} is 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 (5-8):

$$L_S^i = \sum_{k=0}^{f_i'} LM_S^k \tag{5-8}$$

where $f_i'(i = 0, \dots, n)$ 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_S following the given distribution.

The sample set of the total loss L_T can be generated based on simulation work above using formula (5-9):

$$L_T^i = L_P^i + L_S^i \tag{5-9}$$

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 4-1.

5.2.2 Accuracy Evaluation

We evaluate the accuracy of the FAIR model by comparing marginal probability distributions produced by the FAIR model 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 measurement that we use is based on *K-L* (Kullback-Leibler) divergence, which measures the relative entropy from distribution $q(x)$ to $p(x)$, is shown by formula (5-10) [113]:

$$\mathrm{K}(p \parallel q) = \int p(x) ln \frac{p(x)}{q(x)} dx \tag{5-10}$$

Since $K(p \parallel q)$ is not a symmetric measurement, instead we use a symmetric divergence measure referred to as J divergence shown by formula (5-11) [109] [110] :

$$J(p,q) = K(p \parallel q) + K(q \parallel p) \tag{5-11}$$

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 the J divergence the model has against the FAIR-MC, the more accurate the model is.

5.3 Experimental Analysis: Evaluation of FAIR and FAIR-BN

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.

5.3.1 The Design of the Experiments

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 5.3.2, 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 4-1.

In Section 5.3.3 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) [104]. APT can make the targeted information asset dormant under attacks for a long time period. For this reason, using right-long-tailed distributions [103] [102], 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.

The results of the FAIR model are generated using the Open Fair™ Risk Analysis Tool [101], which is built using Excel. Its method of calculation is described in [100]. We have carefully analysed this and have provided more detailed explanation in Chapter 4. We have used AgenaRisk [68], 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 5.1.

We use Matlab [114] 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 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.

5.3.2 Experimental Tests Complying with Assumptions of the FAIR Model

5.3.2-a) Experimental Tests of Risk Aggregation Processes

With the assumptions of the FAIR model, LM_P follows a triangular distribution, $Triangular\ (min = 100, ml = 175, max = 200)$, 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 5-2.

Table 5-2 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	13795	14339	13800	0.0174	0.0161
2	0	230	300	28504	27857	27964	1.0E+08	1.2E+08	1.1E+08	46787	49883	46792	0.0332	0.0323
3	20	80	180	14946	14730	14778	2.8E+07	3.3E+07	3.0E+07	27641	28643	27537	0.0230	0.0195
4	60	250	400	37986	37458	37473	1.2E+08	1.4E+08	1.3E+08	60939	63143	61108	0.0239	0.0354
5	20	250	630	48359	47237	47517	3.8E+08	3.8E+08	4.0E+08	92552	93069	93838	0.0194	0.0189
6	15	30	250	15844	15686	15560	7.2E+07	7.1E+07	7.5E+07	36490	36500	36782	0.0168	0.0160
7	15	30	540	31587	31168	30890	3.6E+08	3.9E+08	3.8E+08	77924	78953	78070	0.0312	0.0225
												Average:	0.0236	0.0230

We also use Euclidean distance [115] and K-L divergence 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 addition, we use *K-L*(FAIR-MC || FAIR) and

K-L(FAIR-MC || FAIR-BN) to represent K-L divergence from FAIR and FAIR-BN to FAIR-MC respectively.

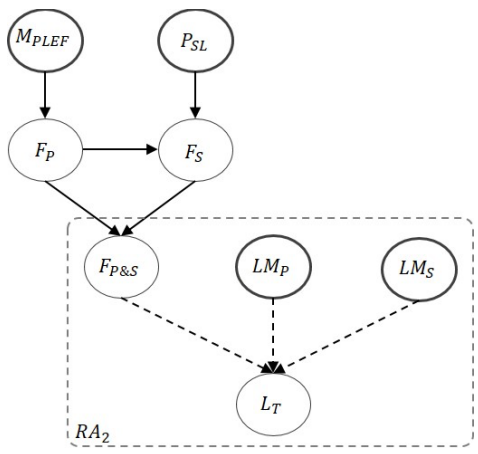
In the seven tests recorded in Table 5-2, the average *Eu(FAIR, FAIR-MC)* vs *Eu(FAIR-BN, FAIR-MC)* is 0.0283 vs 0.0232 while the average K-*L(FAIR-MC || FAIR)* vs *K-L(FAIR-MC || FAIR-BN)* is 0.0122 vs 0.0110. More detailed results are shown in Table 5-3. This result confirms that the three models provide consistent results in these seven tests.

Table 5-3 Results comparison of L_P distributions
with inputs following triangular distributions-with the other measurements

Test	MPLEF			J(FAIR, FAIR-MC)	J(BN, FAIR-MC)	Eu(FAIR, FAIR-MC)	Eu(FAIR-BN, FAIR-MC)	K-L(FAIR-MC ||FAIR)	K-L(FAIR-MC ||FAIR-BN)
	min	mid	max						
1	0	20	90	0.0174	0.0161	0.0232	0.0202	0.0091	0.0079
2	0	230	300	0.0332	0.0323	0.0387	0.0224	0.0126	0.0120
3	20	80	180	0.0230	0.0195	0.0306	0.0213	0.0104	0.0091
4	60	250	400	0.0239	0.0354	0.0339	0.0355	0.0134	0.0158
5	20	250	630	0.0194	0.0189	0.0241	0.0211	0.0086	0.0064
6	15	30	250	0.0168	0.0160	0.0216	0.0203	0.0072	0.0067
7	15	30	540	0.0312	0.0225	0.0263	0.0218	0.0244	0.0194
			Average:	0.0236	0.0230	0.0283	0.0232	0.0122	0.0110

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 Figure 5-8, where boldly outlined nodes represent input variables that are specified using triangular distributions in the FAIR model.

Figure 5-8 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 5-4.

Table 5-4 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 skew	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

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 and the average *K-L(FAIR-MC || FAIR)* vs *K-L(FAIR-MC || FAIR-BN)* is 0.0161 vs 0.0101, which confirms that the three models provide consistent results in these five tests. More detailed results of different measurements are shown in Tables 5-5.

Table 5-5 Results comparison of L_T distributions
-with the other measurements

Test	MPLEF			J(FAIR, FAIR-MC)	J(BN, FAIR-MC)	Eu(FAIR, FAIR-MC)	Eu(FAIR-BN, FAIR-MC)	K-L(FAIR-MC ||FAIR)	K-L(FAIR-MC ||FAIR-BN)
	min	mid	max						
1	0	200	500	0.0362	0.0119	0.0361	0.0128	0.0190	0.0081
2	50	200	1000	0.0237	0.0122	0.0448	0.0160	0.0134	0.0091
3	20	80	200	0.0320	0.0214	0.0331	0.0304	0.0195	0.0137
4	20	160	200	0.0264	0.0202	0.0415	0.0149	0.0148	0.0112
5	0	200	1000	0.0237	0.0145	0.0337	0.0151	0.0139	0.0086
			Average	0.0284	0.0160	0.0378	0.0178	0.0161	0.0101

5.3.2-b) 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 that are implemented by 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): $M_{PLEF} = F_{TE} * P_V$. For this, the average *J*-divergence of the FAIR model and FAIR-BN against the FAIR-MC results are 0.0310 VS 0.0069, which shows the FAIR-BN is more accurate. More detailed results are represented in Table 5-6.

Table 5-6 Comparison results of $M_{PLEF} = F_{TE} * P_V$
With P_V simulated by *Triangular* (0.2, 0.3, 0.7). The calculation is the same as $F_{TE} = F_C * P_A$

Test	Description	FTE			Mean			Variance			99th			J(FAIR,MC)	J(BN,MC)
		min	mid	max	FAIR	BN	MC	FAIR	BN	MC	FAIR	BN	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

2) The Primary Loss Event Frequency (PLEF) is derived from MPLEF following $F_P = P\text{oisson}\ (M_{PLEF})$: here the average *J*-divergence results for FAIR and FAIR-BN against FAIR-MC are 0.0170 VS 0.0059, which shows the FAIR-BN is closer to the standard. More detailed results are represented in Table 5-7.

Table 5-7 Comparison results of F_P = Poisson (M_{PLEF})

Test	Description	MPLEF			Mean			Variance			99th			J(FAIR,MC)	J(BN,MC)
		min	mid	max	FAIR	BN	MC	FAIR	BN	MC	FAIR	BN	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

3) The Secondary Loss Event Frequency (SLEF) is computed from PLEF and Chance of Secondary Loss (CSL) following $F_S =$

$Binomial\,(F_P,\ P_{SL})$ and this produces an average *J*-divergence for FAIR and FAIR-BN against FAIR-MC of 0.0213 VS 0.0053, so again the FAIR-BN model is more accurate. More detailed results are represented in Table 5-8.

Table 5-8 Comparison results of $F_S = Binomial\,(F_P,\ P_{SL})$
With P_{SL} simulated by *Triangular* (0.2, 0.3, 0.7).

Test	Description	MPLEF			Mean			Variance			99th			J(FAIR,MC)	J(BN,MC)
		min	mid	max	FAIR	BN	MC	FAIR	BN	MC	FAIR	BN	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

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 and FAIR-BN have similar performance that each of them wins one time and they produce the same results compared with the FAIR-MC for the rest three tests. More detailed results are represented in Table 5-9.

Table 5-9 Results of $P_V = P(P_{TC} > P_{RS})$
With P_{RS} simulated by *Triangular* (0.2, 0.3, 0.7).

Test	description	PTC			PV		
		min	mid	max	FAIR	BN	MC
1	Include 0	0.0	0.2	0.5	0.17	0.14	0.14
2	Long tail	0.1	0.2	1.0	0.51	0.52	0.52
3	left skew	0.2	0.4	0.8	0.62	0.65	0.65
4	right skew	0.2	0.6	0.8	0.75	0.78	0.79
5	0 and long tail	0.0	0.2	1.0	0.46	0.47	0.46

5.3.3 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).

5.3.3-a) LEF Follows Long-Tailed Distributions

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

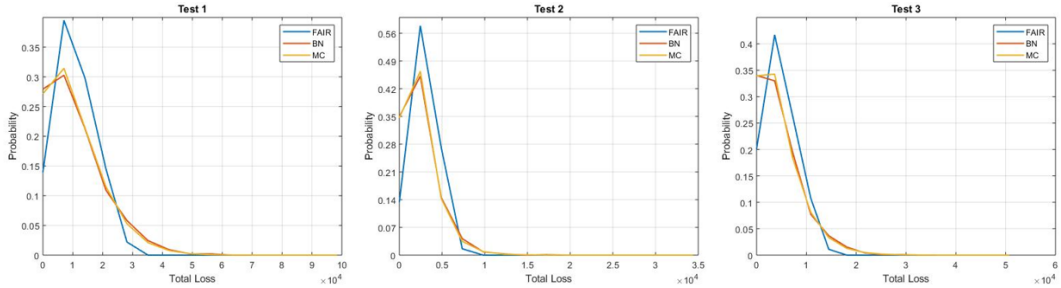
- Weibull distribution (shape = 1.5, scale = 100)
- Log Normal distribution (mean = 3, standard deviation = 0.5)
- Gamma distribution (alpha = 2, beta = 20)

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 5-10. We compare distributions of primary losses, L_P, generated by these three models in Figure 5-9.

Table 5-10 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	29705	43254	42892	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	10034	10218	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	14275	20927	20663	0.5102	**0.0073**
											Average:	0.6066	**0.0103**

Figure 5-9 Results comparison of L_P distributions
with F_P following long-tailed distributions



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 5-2. 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 Figure 5-9. We also use Euclidean distance and K-L divergence as alternative measurements in this test group. The results can lead to the consistent conclusion. More detailed results of this are given in Table 5-11.

Table 5-11 Results comparison of L_P distributions
with F_P following long-tail distributions-with the other measurements

Test	Input Distributions		J(FAIR, FAIR-MC)	J(BN, FAIR-MC)	Eu(FAIR, FAIR-MC)	Eu(FAIR-BN, FAIR-MC)	K-L(FAIR-MC ||FAIR)	K-L(FAIR-MC ||FAIR-BN)
	PLEF	PLM						
1	Weibull	LogNormal	0.7683	**0.0074**	0.1822	**0.0166**	0.6580	**0.0053**
2	Log Normal	LogNormal	0.5412	**0.0161**	0.2745	**0.0147**	0.3785	**0.0136**
3	Gamma	LogNormal	0.5102	**0.0073**	0.1799	**0.0177**	0.4229	**0.0056**
		Average:	0.6066	**0.0103**	0.2122	**0.0163**	0.4865	**0.0082**

The measurements here, including *J*-divergence, Euclidean distance, and the *K-L* divergence, are used as the indicator that can directly reflect the accuracy of the FAIR model and the FAIR-BN. The statistics (mean, variance and 99 quantile) that we have also provided for each experiment would have more realistic meaning. Table 5-10 compares the assessment results of the primary loss that an organization might encounter. We can find that obvious gap, between statistics provided by the FAIR against its peer FAIR-BN and the standard, FAIR-MC, exists in these test cases. More precisely, towards a certain information asset, if the asset suffers from APT, the potential primary loss has been assessed by the three methods. In the FAIR-MC results (i.e., in test 3 from Table 5-10), the mean and 99th quantile values of the primary loss are estimated as 6100 and 20663, while they are estimated as 6400 and 14275 by the FAIR model. This level of difference in predicting the most likely losses and the worst-case losses can obviously influence the budget-restricted

decision making in practice and the influence could be vary from different criteria (introduced in section 2.1.3) adopted by organizations.

5.3.3-b) LEF and LM Using Other Statistical Distributions

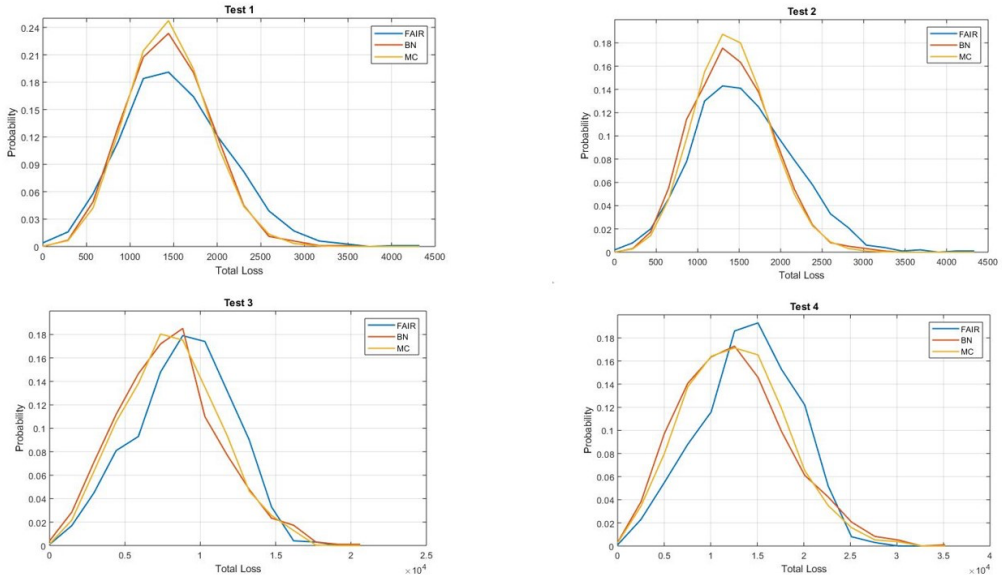
In addition to the long tail distribution scenario, there are other situations that may require different distributions rather than those assumed by FAIR. For example, FAIR uses a Poisson distribution, with an input triangular distribution, to simulate the LEF for the further risk aggregation. In practice, 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 5-12 and Figure 5-10 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 subsection 5.3.2.

Table 5-12 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	15811	16720	16437	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	25186	28721	27864	0.2561	**0.0531**
											Average:	0.2227	**0.0354**

Figure 5-10 Results comparison of L_P distributions
with F_P and LM_P following other distributions



The average *J*(*FAIR-BN, FAIR-MC*) in this test group is 0.0354 which is consistent with the general cases shown in Table 5-2. 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 5-12 and distribution comparisons shown in Figure 5-10, demonstrate the insufficiency of FAIR in the RA_1 process when it approximates distributions of input variables using triangular distributions.

5.3.3-c) 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 5-13, 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 FAIR and FAIR-MC (0.0183 vs 0.0768), leading to the conclusion that the FAIR-BN model is more accurate in this scenario.

Table 5-13 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**

5.3.4 Summary of 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.

5.4 Discussion and Summary

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 subsection 5.3.3, when LEF has the long-tail feature or LEF and LM follow other distributions, the FAIR accuracy decreases. In comparison, FAIR-BN can provide stable 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 4-1, 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.

To make fair comparison on computational efficiency of the three model, we divide the whole calculation through the FAIR model into two parts: 1) calculating LEF (Loss Event Frequency) and its sub-factors 2) risk aggregation processes, RA_1 and RA_2. In part 1, FAIR-BN and FAIR-MC have identical efficiency compared with the FAIR model (the calculation can be finished within 10 seconds). In part 2, the FAIR model is more efficient, that the calculation can be finished within 1 minute, since it uses cached data which is calculated from $27 \times 12 \times 1000000$ samples that are generated by MC simulation [116]. We have tested in our machine that this process needs roughly 30 minutes. In RA_1, FAIR-BN can still have comparable efficiency (within 1 minute), while it requires more calculation time in conducting the RA_2 process (roughly within 40 minutes in average) because each calculation is done anew for each case rather than reused from a cache. Hence, taking consider of both data pre-processing and operational time, the 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.

Moreover, FAIR-BN and FAIR-MC can get benefits from more available computational source (i.e., Using GPU Clusters) and the optimized code efficiency. In comparison, the calculation efficiency of the FAIR model would not be

influenced too much since it has got high calculation efficiency on the operational phase and has decreased the requirements for computation source by using cached data and statistical approximation technique. Hence, with more computational source available and optimizing the code efficiency, the gap between models 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, 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 assigned 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 illustrate the expandability of FAIR-BN by extending it using a process-oriented model and a defend-attack game model in Chapter 8.

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.

In summary, 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.

Chapter 6 Adversarial Risk Analysis and the Bayesian Network Based Implementation

Cybersecurity risk can be regarded as a function of the interplay between the defender (the organisation) and the attacker: decisions and actions made by the defender 'second guess' the decisions and actions taken by the attacker and vice versa. Insight into this 'game' between these two agents provides a means for the defender to identify and make optimal decisions, which is a technical perspective risk analysis. To date, the Adversarial Risk Analysis (ARA) framework has provided a decision-analytical approach to solve such game problems in the presence of uncertainty and uses Monte Carlo simulation to calculate and identify optimal decisions. We propose an alternative framework to construct and solve a serial of sequential Defend-Attack models, that incorporates the adversarial risk analysis approach, but uses a new class of influence diagrams algorithm, called hybrid Bayesian network inference, to identify optimal decision strategies. In this chapter, we use an example to intuitively illustrate how the proposed framework can be implemented. We formally introduce the algorithm of the proposed approach and how it can be adopted for analysing more complex problems in the next chapter. Compared to Monte Carlo simulation, the proposed hybrid Bayesian network inference is more versatile because it provides an automated way to compute hybrid Defend-Attack models and extends their use to involve mixtures of continuous and discrete variables, of any kind. More importantly, the hybrid Bayesian network approach is novel in that it supports dynamic decision making whereby new real-time observations can update the Defend-Attack model in practice. We also extend the Defend-Attack model to support cases involving extra variables and longer decision sequence. Examples are presented, illustrating how the proposed framework can be adjusted for more complicated scenarios, including dynamic decision making.

This chapter provides the introduction of ARA in section 6.1 and illustrate the calculation mechanism of ARA focusing on a typical game model, sequential Defend-Attack (D-A) models, for it can properly represent realistic cybersecurity

cases, in section 6.2. We propose an alternative framework, based on HBN inference and decision trees, to solve the typical sequential D-A games from the ARA perspective. This is represented in Section 6.3, in which we show how to depict the standard D-A game using HBN and Section 6.4, in which we illustrate how to conduct the calculation and support the decision making. We illustrate how to use this framework to solve more complicated D-A problems in Chapter 7. The technologies adopted in the work represented in Chapter 6 and Chapter 7, including influence diagrams and decision trees has been introduced in section 3.4.

6.1 Introduction of ARA

Game-theoretical approaches have been the typical choice to model interplay between two or more strategic adversaries and have been widely applied to cybersecurity issues [12-15]. However, conventional game theory faces a challenge when it aims to find solutions for all the participants of the game, in that the solution must be a Nash equilibrium. A Nash equilibrium is a solution concept that describes a steady state condition of the game. It represents a strategy set of all involved players, in which no player would prefer to change his strategy as that would lower the payoffs given that all other players are adhering to the prescribed strategy [14]. As the problem and associated game models get more realistic and complex, the requirement of Nash equilibrium makes it increasingly difficult to compute a solution [117] [118]. Moreover, it is stated in [105] that the computation of traditional game theory solutions hold the “common knowledge” assumption, that the utilities and beliefs of each participants for a chance node are common (identical) knowledge. However, the participants may have different beliefs of a chance node and the utilities in realistic cases [105]. Therefore, this common knowledge assumption is not adequate to the strategic and behavioural complexity of many real-world applications, especially in the cybersecurity context [105].

Adversarial Risk Analysis (ARA) [119] was proposed to address the above mentioned shortcomings of classic game theory. ARA uses probabilities to describe the decision maker’s (typically the defender’s) subjective beliefs, anticipating the opponent’s potential decisions and actions, and by this way providing an alternative

solution for uncertainty in classic game theory [119]. In ARA, participants (i.e., the defender and the attacker) can have different utilities and different beliefs about a chance node, since the decision problem is divided by the certain problems corresponding to each participant. For example, the defend-attack problem can be solved by tackling with the attacker's problem and the defender's problem (represented by Figure 6-1 and Figure 6-2 and will be explained in the next section). In the individual problems, a chance node can be modelled based on different belief from the view of the corresponding participant. Using the common knowledge of chance nodes can be regarded as a simplified version of this.

General security risk analysis problems, as explored in [120] [121] [122], are modelled as a number of basic templates (i.e. simultaneous D-A model, sequential D-A model, etc) with a known ARA solution in [105]. The templates differ in the way and order in which the attack and defence actions take place within the global sequence of decisions and events, as well as in the information revealed. These templates can then be represented by Influence Diagrams (ID) [37].

How to best model and efficiently calculate optimal decisions using ARA has received a lot of attention in recent years. Opponents in simultaneous decision making games are modelled following ARA in [123]. Insider threat in sequential D-A games were modelled using the ARA approach in [117]. A calculation procedure for conducting ARA for a bi-agent game is introduced in [124]. In the work [124], a model consists of sequential D-A pattern and simultaneous D-A pattern is considered. For more practical cases, [125] provides an ARA framework for cybersecurity risk analyse using insurance as part of the security portfolio for decision making and the work done by [126] applied ARA in Counterterrorist Online Surveillance.

It is argued in [117], that in most realistic cybersecurity cases, the defender would deploy their defence first to deter and prevent attacks and, therefore, it makes sense to model the cybersecurity problem as a sequential D-A game, rather than as a simultaneous one. However, solving the sequential D-A model, and its more challenging extensions i.e., the sequential D-A model with extra variables or with

a longer decision sequence, has not been systemically investigated in previous research. In this thesis, we focus on solving the bi-agent sequential D-A game model and its extensions. We provide a Hybrid Bayesian Network (HBN) based ARA approach as a comprehensive solution and use examples to illustrate how the proposed framework can be applied to practical problems. Our proposed solution can be easily applied to solve other typical sequential game templates summarized in [105]. For example, the D-A-D model can be regarded as an instance of the sequential D-A model with longer decision sequences. Moreover, solving sequential A-D models [105] and the extensions (i.e., sequential A-D-A models) can be regarded as a reflective solution to the "dual problem" of solving the D-A problem, since only the order of making decision changes while the underlying calculating mechanism remains the same compared with solving the D-A model.

Most ARA solutions use Monte Carlo (MC) simulation to carry out the calculation, for example in [125] [123] [117] [124]. MC simulation is straightforward to implement. However, this approach can become computationally challenging when dealing with decision dependent uncertainties, especially in D-A models where we encounter longer decision sequences. Moreover, it cannot cope with new evidence that could be used to update the game model, dynamically, in real time, which we contend is a realistic requirement for practical use. In our work, we provide algorithms to implement the ARA approach based on the HBN inference [64]. The proposed method offers a fully automated way to compute sequential D-A models and can support dynamic decision analysis which has not been solved by previous ARA solutions that adopt MC simulation.

6.2 Adversarial Risk Analysis of the Defend-Attack Model

Adversarial Risk Analysis (ARA) provides a decision analytic approach offering prescriptive supporting one of the intervening agents (i.e., the defender) based on an expected utility model treating the adversary's decisions as uncertainties. As we have mentioned, since it is rational to model the cybersecurity problem as a sequential D-A game, rather than as a simultaneous one [117], we focus on solving the sequential D-A model and its extensions in this thesis. Fundamentally, ARA

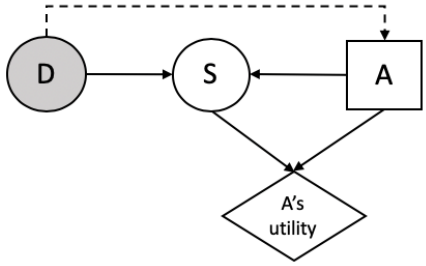
solves the D-A game by analysing the attacker's problem, anticipating his best choice and taking into account the defender's own options for the most optimal defence strategy. In the following content of this chapter, we formally illustrate how to use the HBN to solve the typical D-A game from the ARA perspective.

The adversarial risk analysis of the sequential Defend-Attack (D-A) game model provides a template and procedure to identify the optimal strategy for the defender. In this section, we analyse the D-A game represented by Figure 3-3 from the ARA perspective and illustrate how to construct an HBN for the calculation and supporting decision making for the defender.

We assume that in the ID represented by Figure 3-3, the defender has a discrete set of possible defence levels, which are represented by the decision node D (Defences). After observing the potential defence levels that can be implemented, the attacker creates a discrete set of possible attack levels $A = \{a_1,\ a_2, \dots, a_m\}$ represented by node A (Attacks). A dashed arc pointing from node D *t*o A represents the fact that the attacker's decision depends on the potential defence. From the ARA perspective, the D-A game can be divided into the attacker's problem and the defender's problem.

To determine the defender's best choice, the defender would analyse the attacker's problem first, which is represented by Figure 6-1, to anticipate choices the attacker might optimally make. Then, based on this analysis, the defender would determine the optimal strategy for herself in the first place. This calculation procedure is an implementation of backwards induction [127] [105]. Backwards induction analyses decisions from the end to the beginning of the decision sequence to calculate optimal strategies for decision nodes. Assuming rationality, the attacker should choose the strategy that can maximize his utility, given all the potential defence choices, and that the defender will take this into account.

Figure 6-1 The attacker's problem in the D-A game



$A's$ expected utility corresponding to each possible combination of $(d, a) \in D \times A$ is:

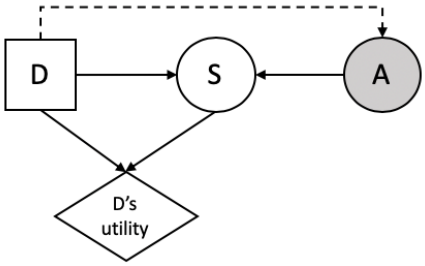
$$\Psi_A(d,a) = p_A(S=0|d,a)u_A(d,a,S=0) + p_A(S=1|d,a)u_A(d,a,S=1) \quad (6\text{-}1)$$

Therefore, the defender can predict that the optimal attack that would be adopted is:

$$a^*(d) = argmax_{a\in A}\Psi_A(d,a), \forall d \in D \quad (6\text{-}2)$$

Consequently, the defender can calculate her optimal initial strategy to adopt in the game by analysing the defender's problem which is shown in Figure 6-2.

Figure 6-2 The defender's problem in the D-A game



The expected utility of D corresponding to each possible combination of $(d, a) \in D \times A$ is:

$$\Psi_D(d,a) = p_D(S=0|d,a)u_D(d,a,S=0) + p_D(S=1|d,a)u_D(d,a,S=1) \quad (6\text{-}3)$$

Under the assumption that the opponent in this game is rational, her best choice is:

$$d^* = argmax_{d\in D}\Psi_D\big(d, a^*(d)\big) \quad (6\text{-}4)$$

This calculation follows backwards induction as is represented that, the decision sequence in reality is from D to A, while the analysing/calculating sequence is backwards, from A to D.

Note that, in contrast with classic game theory, the solution d^* for the sequential game need not correspond to a Nash equilibrium, since in ARA, players are not assumed to have full and common knowledge and the solution d^* is derived from the predicted (p_A, u_A) rather than the actual one [105].

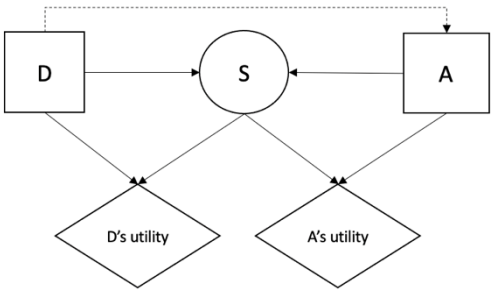
6.3 Depicting the Defend-Attack Game Problem Using Hybrid Bayesian Networks

Here we use an example to show how to implement the sequential D-A game using a Hybrid Bayesian Network (HBN), that models a concrete sequential Defend-Attack game, as shown in Figure 6-3 (a). We simplify the opponents' decisions as Boolean variables representing to defend or not, for the defender, and to attack or not, for the attacker. We assume that the defender's decision is about whether to defend an information asset. Meanwhile, after observing whether the defender defends, the attacker would consider whether to attack. We show the setting of nodes' CPTs in Figure 6-3 (b).

We assign uniform distributions to the decision nodes: node D (defence decision) and node A (attack decision) representing the opponents' open-mindedness choices. The CPT of the Success node (node S) models how the attack and defence interact to determine the probability of a successful attack. The node S can be true or false. If an attack is not made, the probability of S to be true is zero. We assume that if an attack is made, the probability of success is 0.8 if the asset is undefended, while it decreases to 0.2 if defended. The utility node, $D's$ Utility, models the defender's payoff given the asset is defended (utility: -100) and the cost to the defender of a successful attack is (utility: -200). The defender can predict the attacker's utility based on the assumption that the attacker's attack cost (utility: -100) and the payoff from a successful attack (utility: +200).

Figure 6-3 The BN model of a sequential D-A game

(a)



(b)

CPT for the node D:

No	0.5
Yes	0.5

CPT for the node A:

D	No	Yes
No	0.5	0.5
Yes	0.5	0.5

CPT for the node S:

D	No		Yes	
A	No	Yes	No	Yes
False	1.0	0.2	1.0	0.8
True	0.0	0.8	0.0	0.2

CPT for the node D's Utility:

D	No		Yes	
S	False	True	False	True
Expressions	Arithmetic(0)	Arithmetic(-200)	Arithmetic(-100)	Arithmetic(-300)

CPT for the node A's Utility:

S	False		True	
A	No	Yes	No	Yes
Expressions	Arithmetic(0)	Arithmetic(-100)	Arithmetic(0)	Arithmetic(100)

6.4 Risk Assessment and Decision Support for the Defender

In section 6.2, backwards induction is introduced to determine the optimal strategy for the defender in the D-A model in general. In this subsection, we illustrate how to implement the backwards induction for calculating the optimal decision for the defender in the HBN shown in Figure 6-3. To achieve this, there are three steps involved.

Firstly, the defender would initially analyse the attacker's problem, as shown in Figure 6-1, to predict what attacks he might make against possible defences. At this point, we regard the defender decision choices as a variable that might be potentially observed by the attacker and used to inform his decision making. Since the decision node for the defender becomes a chance node in this subproblem, we use an oval node to represent it. Here, the attacker's judgment is that there is a fifty-fifty chance of the defender defending or not. We calculate the attacker's utility of attacking, or not, under the two scenarios and identify those choices that maximize his utility, given he observes the defender's action. This calculation follows formula (6-1). In the Influence Diagram Analysis Function in AgenaRisk, this calculation can be done automatically by selecting node A to be the decision node, node D and S to be the chance node and node $A's$ Utility to be the utility node [128]. In this

example, we aim to maximize the expected utility and the calculation results are graphically represented by the Decision Tree (DT) shown in Figure 6-4.

Figure 6-4 The DT of the attacker's problem

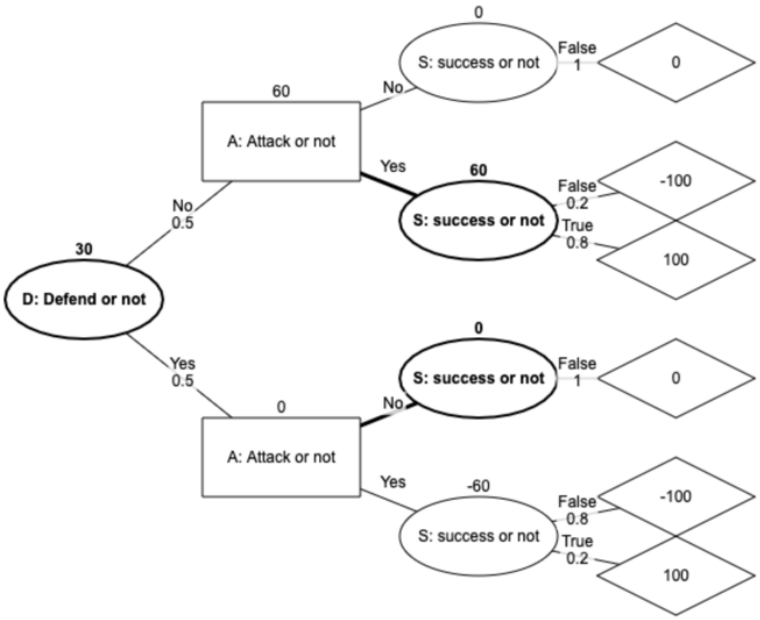
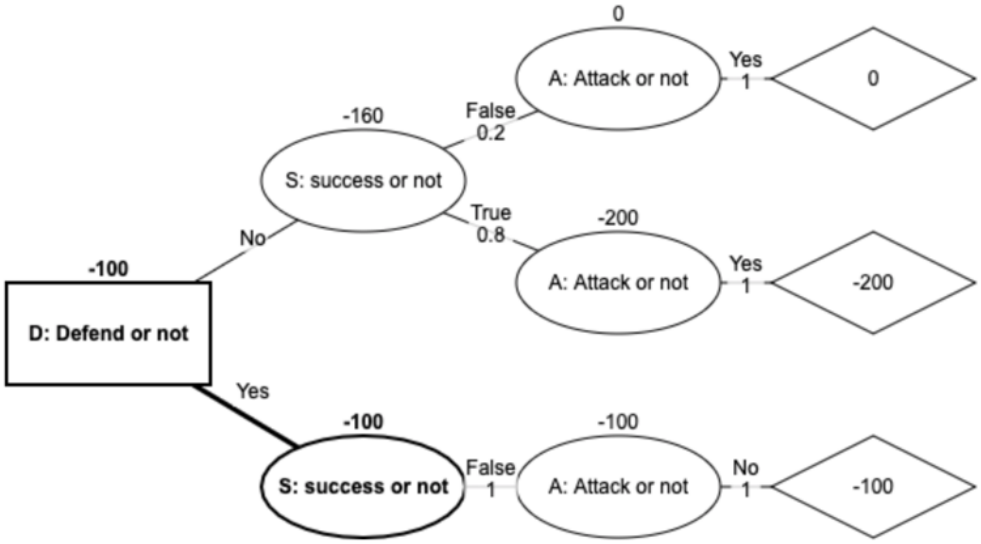


Figure 6-4 shows the best decision for the attacker in bold arcs, occur when the defender does not defend herself, and the best choice for the attack is to attack, which provides him the maximum utility (60), while if the defender defends, the attacker's best choice would be to not attack, with the maximum utility (0).

The second step is to update the CPT of the attacker's decision, A, in the model. The CPT for node A is not 0.5 vs 0.5 anymore. The updated CPT for A is $P(A = No|D = Yes) = 1,\ P(A = Yes|D = Yes) = 0,\ P(A = No|D = No) = 0$ and $P(A = Yes\ |D = No) = 1$ according to the results represented by Figure 6-4.

Then, in the last step, we determine the defender's optimal decision by analysing the defender's problem (shown in Figure 6-2) using the updated D-A model. In this step, the decision of the attacker is regarded as a variable dependent on the defender's decision. Hence, node A becomes a chance node in the defender's problem. We calculate the defender's utility of defending or not following formula (6-3). We choose node D as the decision node, node A and S to be the chance nodes and node $D's$ Utility as the utility node. The calculation results are graphically represented by the DT shown in Figure 6-5.

Figure 6-5 The DT of the defender's problem



Here the optimal choice for the defender is shown by the bold arc, where to maximize her utility (-100) she should defend, otherwise she would suffer from a worse expected payoff (-160) if she does not. Therefore, the optimal decision for the defender is $D = Yes$ and sequentially, the attacker is anticipated that might not conduct the attack ($A = No$).

Chapter 7 Advanced Sequential Defend-Attack Game Model

The work represented in this chapter includes: 1) we illustrate how to use the proposed framework introduced in Chapter 6 to solve more practical D-A problems involving extra variables and longer decision sequence (also known as multi-period game/ k-level thinking); 2) we present numerical examples to show how our framework can support decision making in different application contexts involving extra variables, longer decision sequences and dynamic decision making.

In this chapter, we summarize the rules needed to be followed to extend D-A models for more complicated scenarios, i.e., D-A problems with extra variables and with longer decision sequences and apply these rules to two examples in section 7.1. Consequently, we discuss dynamic decision analysis, provide the algorithm and illustrative examples in section 7.2. A summary is presented in section 7.3.

7.1 Extensions of the Sequential Defend-Attack Game Model

In Chapter 6, we describe how to implement influence diagrams of the D-A game model using Hybrid Bayesian Networks (HBNs) and consequently how to conduct the calculation. To illustrate the calculation mechanism, we build models with core variables only, comprising decision nodes, the chance nodes representing if the attacks are successful, or not, and the utility nodes for the two agents. However, in practice, interaction between defenders and attackers may involves more factors [105]. In this section, we summarize the rules required when building and calculating more complicated sequential D-A game models.

7.1.1 Rules to Build and Calculate the Sequential Defend-Attack Game Models

Extending the sequential D-A game models with extra variables or longer sequences is feasible, so long as we follow these rules:

1) Decisions from D and A need to be made alternately. This is described as *level-k thinking* in [105]. For example, a D-A-D model can be considered as the defender, the attacker, then the defender decides. In this way, (local) optimal decision for each decision phase (obtained using backwards induction) can lead to the global optimal decision set for the whole adversarial problem.
2) Each decision node is influenced by all the previous decision nodes in the sequence representing when making decision on the certain phase, the agent has the knowledge of all the previous decisions. This can be reflected by creating arcs from all its previous decision nodes $1, \dots, i-1$, for each decision node i, pointing to the current node i.
3) Set the chance nodes (i.e., success nodes) and utility nodes following dependent relations below:
   3.1) $S_i = S_i(D_{i-1}, A_i)\ with\ i = \{2, \dots, n\}$;
   3.2) $U_D = U_D(S_2, \dots, S_n, D_1, \dots, D_{n-1})$;
   3.3) $U_A = U_A(S_2, \dots, S_n, A_2, \dots, A_n)$;
4) Decision nodes should be discrete variables to guarantee that each decision is made from finite options.
5) Decision nodes are set to follow uniform distributions to represent open mindedness when making decision. The decision of taking no action is also represented by a state of the decision nodes.

After constructing the sequential D-A model (with extra variables or a longer sequence), we use probabilistic inference in HBNs and constructing Decision Trees (DTs) implementing the backwards induction in the sequential game model to calculate optimal strategies for the defender. In each decision phase in the sequence, we concentrate on the current decision node and regard all its previous decision nodes as chance nodes. The CPT of the current decision node is defined using probabilities conditioned on the adoption of potential strategies given all the combinations of decisions made before the current phase. The initial setting of this CPT is as a uniform distribution representing the agent's open mindedness. We can calculate the agent's utility for each decision option, conditional on each

combination of previous decision, and determine the decision that provides the maximum utility value for the agent. We use this information to update the CPT of the current decision node, following the idea that, given previous decisions, the agent will only adopt decisions that lead to maximum utilities, and hence the probability of the agent adopting any decision other than this must be set to zero. Next, we move to the decision node following the current node and repeat the same operation. When we find the optimal strategy for the first decision node in the sequence, we stop and obtain the optimal strategy set for the defender and the predicted action set anticipated for the attacker with an accompanying maximum utility. We formally summarize this calculation process in Algorithm 1 below:

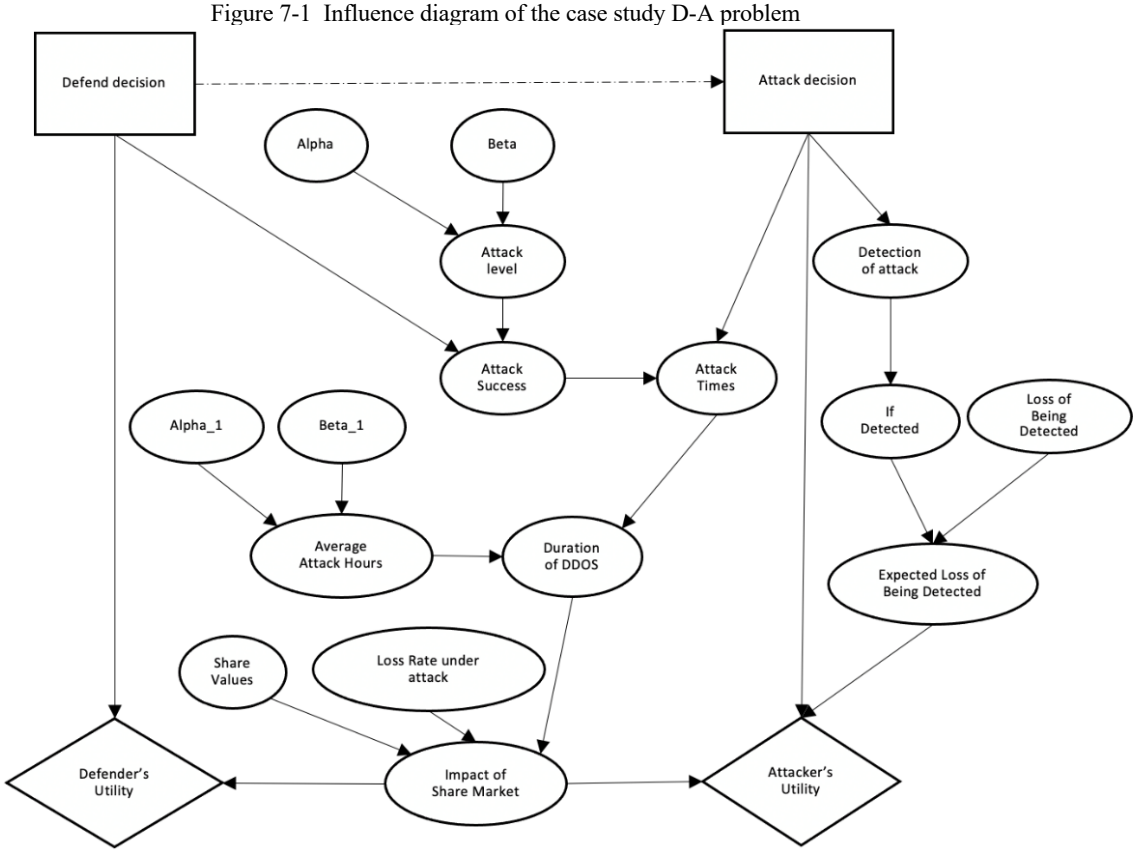
Algorithm 1: The HBN based ARA approach in solving sequential D-A models

Initialization:
$\Psi_A = \Psi_A(D_1, \dots, D_{n-1}, \dots, A_2, \dots, A_n, V)$
$\Psi_D = \Psi_D(D_1, \dots, D_{n-1}, \dots, A_2, \dots, A_n, V)$
(Assume n is even. When n is odds, the calculation follows the same process)
for $(i = n, i > 0, i - -)$ ***do***
if $(i\ is\ odd)$ ***do***
calculate:
$a_i^* = argmax_{a_i \in A_i} \Psi_A(D_1, \dots, d_{i+1}^*, \dots, d_{n-1}^*, A_2, \dots, A_i, \dots, a_n^*, V)$
update the model:
$\Psi_A = \Psi_A(D_1, \dots, d_{i+1}^*, \dots, d_{n-1}^*, A_2, \dots, a_i^*, \dots, a_n^*, V)$
$\Psi_D = \Psi_D(D_1, \dots, d_{i+1}^*, \dots, d_{n-1}^*, A_2, \dots, a_i^*, \dots, a_n^*, V)$
else if $(i\ is\ even)$ ***do***
calculate:
$d_i^* = argmax_{d_i \in D_i} \Psi_D(D_1, \dots, D_i, \dots, d_{n-1}^*, A_2, \dots, a_{i+1}^*, \dots, a_n^*, V)$
update the model:
$\Psi_A = \Psi_A(D_1, \dots, d_i^*, \dots, d_{n-1}^*, A_2, \dots, a_{i+1}^*, \dots, a_n^*, V)$
$\Psi_D = \Psi_D(D_1, \dots, d_i^*, \dots, d_{n-1}^*, A_2, \dots, a_{i+1}^*, \dots, a_n^*, V)$
end for
output:
updated model:
$\Psi_A = \Psi_A(d_1^*, \dots, d_{n-1}^*, a_2^*, \dots, a_n^*, V)$
$\Psi_D = \Psi_D(d_1^*, \dots, d_{n-1}^*, a_2^*, \dots, a_n^*, V)$
Optimal strategies for the defender:
$\{d_1^*, \dots, d_{n-1}^*\}$

7.1.2 Example 1: The Defend-Attack Game with Extra Variables

We now apply the proposed framework to a real cybersecurity problem, a simplified version of the case in [125] [129]. The problem is depicted in the ID shown in Figure 7-1, where the model represents a defender facing a competitor, the attacker, that may attempt a DDoS attack to undermine the availability of the defender's website, compromising her customer services and leading to a decrease in share price. The distributions for modelling the involved variables are specified based on empirical data in [125]. In this example, we comply the identical setting.

Figure 7-1 Influence diagram of the case study D-A problem



The decision node *Defend decision* (D) represents cloud-based DDoS (Distributed Denial of Service) protection (with states 0, 2, 5, 10 and 100 gigabits per second ($gbps$)) that the defender can deploy. The level of defence can be observed by the attacker, and therefore influences the attacker's decision, represented by the node *Attack Decision*. The node *Attack Level (AL)*, which represents the scale of the attack, is assumed to follow a Gamma distribution, with the parameters $Alpha$ and

$Beta$ follow uniform distributions derived from historical data. The defence deployment D and the AL determine the probability of *Attack* Success (AS) together. The variable *Attack Times* (AT) is influenced by both *Attack decision* and AS and is assumed to follow a Binomial distribution. The *Average Attack Hours* (AAH) is assumed to follow a Gamma distribution, of which the parameters $Alpha\_1$ and $Beta\_1$ also follow uniform distributions derived from historical data [125]. The *Duration of DDoS* is derived from AT multiplied by AAH. *Impact of Share Market* is derived from the organization's *Share Value* (i.e., £1500000)*, Loss Rate under Attack* and *Duration of DDoS*. In addition, the Defender's Utility (DU) is influenced by the Defend decision and the *Impact of Share Market*. Different defence deployment incurs different costs (i.e., 2 gbps: £2400; 5 gbps: £3600; 10 gbps: 4800; 100 gbps: £12000) [125]. The defender's utility is equal to the deployment cost plus its loss in the share market.

The attacker needs to decide how many days to conduct the attack over a one-month period. This decision is represented by the node *Attack decision* with state values from 0 to 30. The longer the attack period is, the more likely the attack will be detected, represented by the node *Detection of Attack*, and is assumed to follow a Binomial distribution [125]. If the attack is detected, the attacker would face legal costs, reputational costs, etc, which is represented by *Loss of being Detected*. If the attack is not detected, the *Detection Loss* will be zero. We assume the amount that the defender losses in the share market is the gain of the attacker. Based on that, *Attacker's Utility* is set to equal to *Impact of Share Market* minus *Expected Loss of Being Detected* minus the cost of *Attack Decision*.

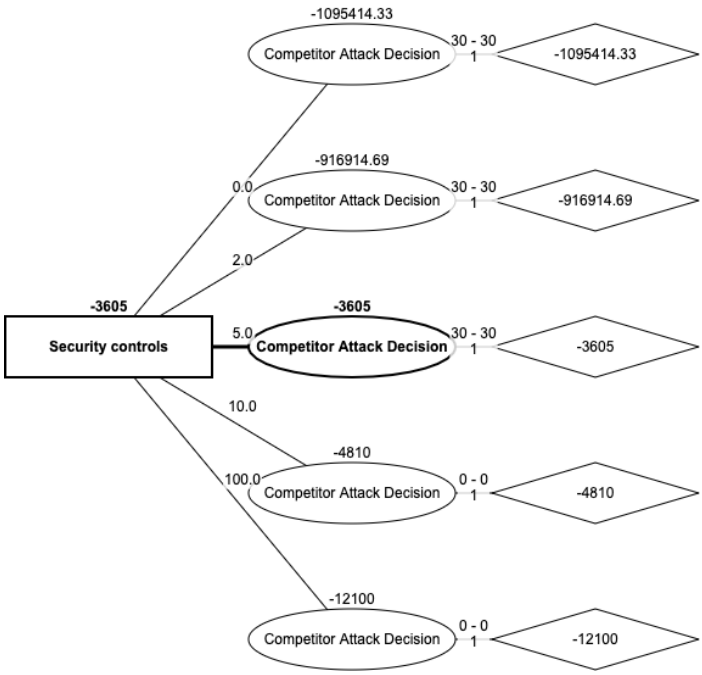
We summarize the variables and how we assign expressions for them in AgenaRisk in Table 7-1.

Table 7-1 Variables and their expressions in Example 1

Variable	Notation	Expression
Defend decision	D	$Uniform\ (D), D = \{0, 2, 5, 10, 100\}$
Attack decision	A	$Uniform\ (0, 30)$
Alpha	a	$Uniform\ (4.8, 5.6)$
Beta	b	$Uniform\ (0.8,1.2)$
Attack Level	AL	$Gamma\ (a, b)$
Attack Success	AS	$min\ (max\ (AL - D, 0.0)/(D + 1.0E - 4),1.0)$
Attack Times	AT	$Binomial\ (number\ of\ trials: A;\ Probability\ of\ success: AS)$
Detection of Attack	DoA	$Binomial\ (number\ of\ trials: A;\ Probability\ of\ success: 0.002)$
If Detected	ID	$if\ (DoA\ > 0.0, "True", "False")$
Loss of Being Detected	LBD	$Normal\ (Mean: 2430000, Variance: 400000)$
Expected Loss of Being Detected	$ELBD$	$\{if\ ID\ =\ False: Arithmetic\ (0.0), if\ ID\ =\ True: Arithmetic\ (LBD)\}$
Alpha_1	$a1$	$Uniform\ (3.6,4.8)$
Beta_1	$b1$	$Uniform\ (0.8,1.2)$
Average Attack Hours	AAH	$Gamma\ (a1, b1)$
Duration of DDoS	DoD	$AAH\ \times AT$
Share Values	SV	1500000
Loss Rate under attack	LR	$Uniform\ (0.00521, 0.00833)$
Impact of Share Market	ISM	$min\ (SV, DoD\ \times\ LR\ \times\ SV)$
Defender's Utility	DU	Partitional expression given status of D: $\{D = 0: -ISM, D = 2: -ISM - 2400, D = 5: -ISM - 3600,$ $D = 10: -ISM - 4800, D = 100: -ISM - 12000\}$
Attacker's Utility	AU	$ISM\ -\ ELBD\ -\ 792.0\ \times\ A$

We conduct calculation using Algorithm 1 with the results as shown in Figure 7-2. This shows the optimal strategy for the defender is $d^* = 5$ based on her analysis of the attacker's problem, of which the optimal attack is predicted to be $a^* = 30$. In this case, the maximum utility of the defender is -3605.

Figure 7-2 Results of the D-A model with extra variables

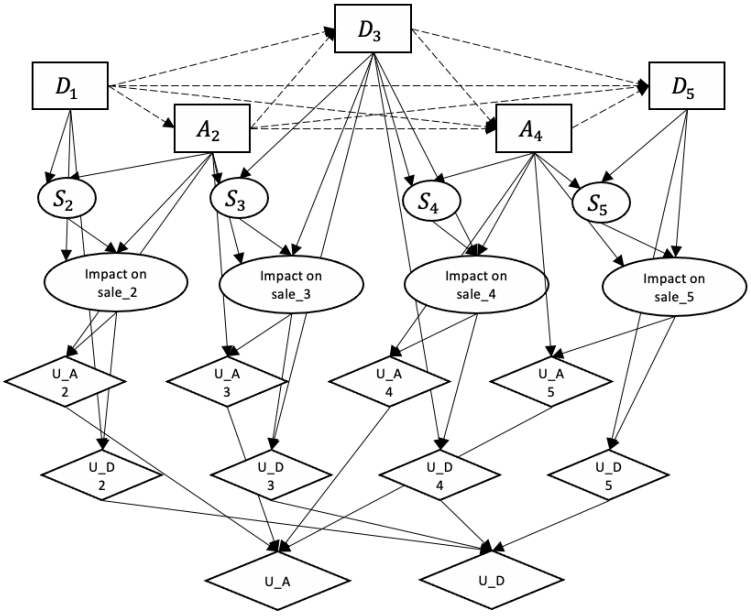


7.1.3 Example 2: The Defend-Attack Game with a Longer Decision Sequence

We now apply the proposed framework to represent and solve a practical cybersecurity problem with more rounds of interaction between the defender and the attacker. We first construct the Influence Diagram (ID) for the game using an HBN, and then illustrate how to apply Algorithm 1 to calculate the optimal strategy for the defender.

The organization provides online services for clients during the working days (Monday to Friday). Its system faces threats from a potential attacker who contemplates a DDoS attack aiming to disrupt the online service provided by the defender. To guarantee the normal operation of the online service, the defender deploys cloud-based protections against attacks. To simplify the problem, we assume the defender can adopt 0, 12, and 24 hours of protection a day, where zero hours means no protection is adopted; 12 hours means protection spreading in the whole day and the total volume is 12 hours; 24 hours is full protection. We also assume that the protection will be deployed when the defender make decision (D_1) on Monday and will remain valid until the next day (Tuesday). For the attacker, after observing defender's deployment, he would make an attack decision on Tuesday (A_2). We assume the attacker has three similar decision choices: conduct a 0, 12 or 24-hour long attack. Similarly, the attack deployment would be valid on the current day through to the next day. When the defender observes the attacker's action on Tuesday, she would make her defence decision (D_3) on Wednesday. This process continues until the weekend. We illustrate this adversarial problem in Figure 7-3.

Figure 7-3 The influence diagram of the Defend-Attack game with longer sequence

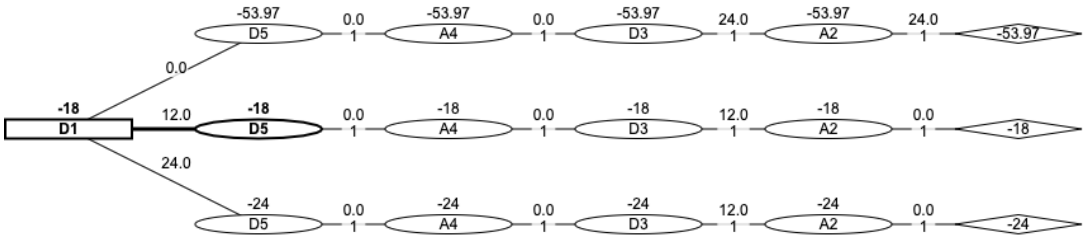


We represent decision nodes for the two agents as D_1, A_2, D_3, A_4 *and* D_5. For D_1, the node has three states (0, 12, 24) with uniform distribution. A_2 is conditional on D_1, and therefore has nine states. Following the same rule, the decision node D_5 has $3^5 = 243$ states. We use S_i $(i = 2, \dots, 5)$ to represent whether the attack is successful. In this case, we set S_i to be Boolean variables that have states "true" and "false". When the attack time exceeds the defence time, some percentage of the online service will be interrupted. $Impact\ on\ sale_{i=2,\dots,5}$ represents on day i, how many service orders are affected by a successful attack. We calculate the percentage of unprotected hours in any day to measure those service orders interrupted by cyber-attack and assume, on average, the defender will have 1000 total online service orders and 10% of those orders may be affected by the cyber-attack. For example, on Tuesday, $A_2 = 12,$ and $D_1 = 0$, then the number of interrupted orders would be $\frac{12-0}{24} \times 1000 \times 10\% = 50$. We use a TNormal distribution to represent the number of orders interrupted by the attack with uncertainty, where we set: 1) mean = $\frac{A_i - D_{i-1}}{24} \times 1000 \times 10\% = \frac{A_i - D_{i-1}}{0.24}$; 2) variance = 400; 3) lower bound = 0; 4) upper bound = 200. Nodes U_{D_i} represent the defender's utility on day i, which is the cost of deploying protection (£500/hour) and the loss caused by interrupted online service (£300/order). For the attacker, we assume his utility is the gain from

the organization's loss on orders minus the cost of conducting attacks (£500/hour), which on day i is represented by the node U_{A_i}. Since utility is additive, we obtain the defender's utility over the whole week, U_D, from $\sum_{i=2}^{5} U\_D_i$. We do the same with the attacker's utility.

We calculate the optimal strategy using Algorithm 1 and the optimal strategy of the subgame in each decision phase can be identified using a Decision Tree representing all the possible decision path and the resulting utility. We determine the optimal decisions and use these to update the model and repeat the process until we identify the optimal strategy at the first decision node in the sequence. The results are shown in Figure 7-4, where the optimal strategy calculated for the defender is $\{d_1^* = 12, d_3^* = 12, d_5^* = 0\}$, while for the attacker the anticipated strategy is $\{a_2^* = 0, a_4^* = 0\}$. In this case, the maximum utility of the defender is -18 (a loss of £18,000).

Figure 7-4 Results of the D-A model with longer sequence



Decision trees constructed in the process are shown in Figure 7-5 ~ Figure 7-8.

Figure 7-5 The DT of D5

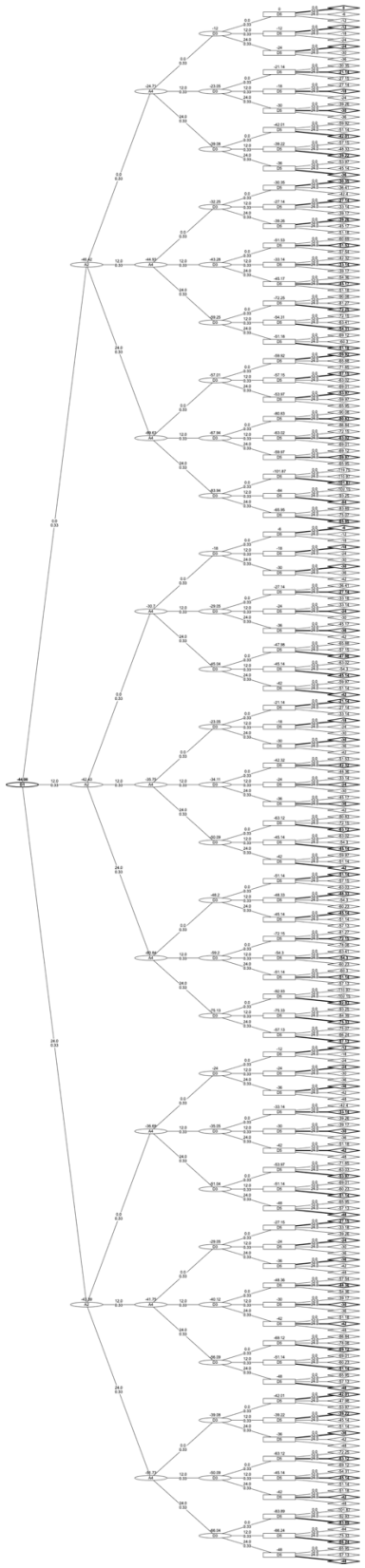


Figure 7-6 The DT of A4

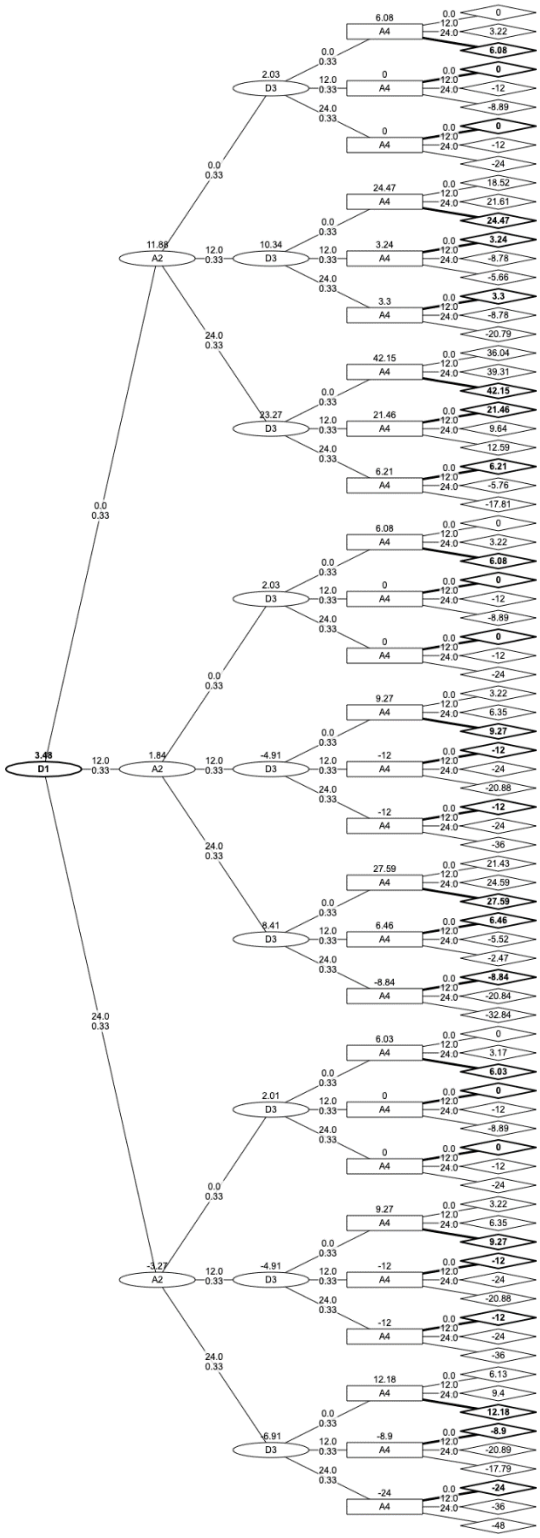


Figure 7-7 The DT of D3

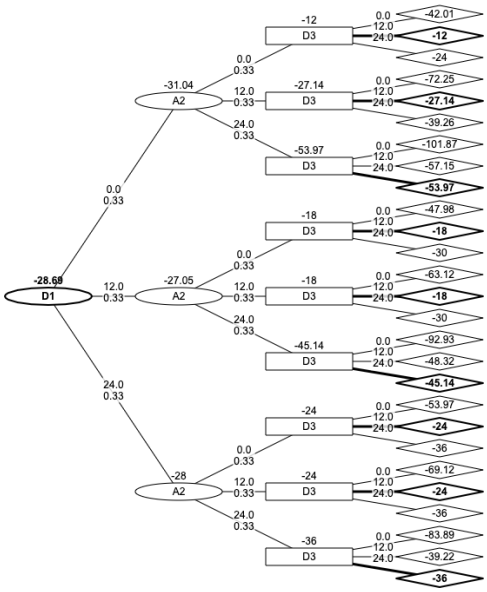
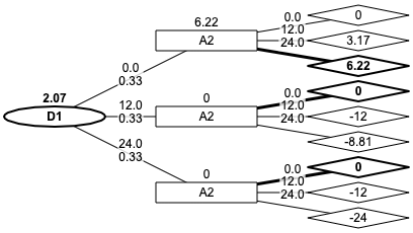


Figure 7-8 The DT of A2



7.2 Supporting the Defender's Dynamic Decision Making

In a game with a longer decision sequence, *k*-level thinking (i.e., he anticipates, she anticipates, what he would anticipates) is involved. As is shown in example 2 (section 7.1.3), the defender anticipates what the attacker would do based on her belief of the attacker's utility (assuming that implementing an attack and a defence costs the same per hour and the loss of the defender caused by attack is the gain of the attacker) and the attack success probability. Based on the analysis of the attacker's problem, the defender determines her optimal decision at the first place. However, the original decision-making problem can become a dynamic decision-making problem over time, meaning that the defender can always use the fresh-observed data to update the model and make real-time-updated decision accordingly. More precisely, in example 2, the calculated strategies for the defender are optimal standing on Monday without observing any other information. When it comes to Wednesday, the defender can make decision based on what she can

observe, i.e., the attack that actually taken or its consequences on Tuesday, and what she can anticipates, i.e., the attack that likely to be conducted in the future (on Thursday). Hence, Dynamic Decision Analysis (DDA), which involves updating the D-A game model using observable information through the decision sequence, is required.

In this section, we provide the DDA algorithm to deal with dynamic decision making. Moreover, we represent two examples to illustrate how the proposed DDA algorithm can be applied to analyse practical problems modelled by HBN.

7.2.1 The Algorithm for Dynamic Decision Analysis

The core idea of supporting dynamic decision making here is to update the sequential D-A model with real-time data and conduct decision analysis on the model updated using this data. We consider this dynamic decision-making issue based on example 2. In the example, we have obtained the optimal strategy set for the defender standing on Monday. This strategy set includes the optimal strategies suggested for the defender on Monday (D_1), Wednesday (D_3) and Friday (D_5) based on her prediction of attacks on Tuesday and Thursday (A_2 and A_4 respectively). Hence, initially, the defender would take the optimum decision d_1^* on Monday. On Wednesday, the defender can then observe effects of the adopted action or its consequence from the previous day, Tuesday in this case. If the actual attack is observable, we use a_2 to represent the observed attack (which does necessarily need to equal to the predicted attack a_2^*). Then we would use the calculated d_1^* and the observed a_2 (rather than the calculated a_2^*) to update the sequential D-A model; otherwise, if only the attack consequence is observable (i.e., whether the attack succeeded on Tuesday), we use this new observation for S_2 to update the model. In the latter case, if the attack on Tuesday is not observable, we remove the arc pointing from node A_2 to D_3. In addition, to represent the fact that D_3 is influenced by S_2, we add an arc pointing from S_2 to D_3. Then we conduct dynamic decision analysis based on the updated model.

We formally summarize the process of supporting dynamic decision-making using Algorithm 2.

Algorithm 2: The HBN based ARA approach in solving dynamic decision making

Initialization:

$\Psi_A = \Psi_A(D_1, \dots, D_{n-1}, \dots, A_2, \dots, A_n, V)$

$\Psi_D = \Psi_D(D_1, \dots, D_{n-1}, \dots, A_2, \dots, A_n, V)$

(Assume n is even. When n is odds, the calculation follows the same process)

for $(j = 1, j \le n-1, j = j+2)$ ***do***

In the latest model conduct ***Algorithm 1*** and get **outputs*:***

The updated model:

$$\Psi_A = \Psi_A(d_1^*, \dots, d_{n-1}^*, a_2^*, \dots, a_n^*, V)$$

$$\Psi_D = \Psi_D(d_1^*, \dots, d_{n-1}^*, a_2^*, \dots, a_n^*, V)$$

Optimal strategy set of the game:

$$\{d_1^*, \dots, d_j^*, \dots, d_{n-1}^*, a_2^*, \dots, a_{j+1}^*, \dots, a_n^*\}$$

identify optima for D_j, which is d_j^*;

if (the actual adopted attack is observable)

then do

record the actual attack adopted in day $j+1$, which is $A_{j+1} = a_{j+1}$;

update the model using $D_j = d_j^*$ and $A_{j+1} = a_{j+1}$;

else if (only the consequence of attack is observable)

then do

record the consequence of attack in day $j+1$, which is $S_{j+1} = s_{j+1}$;

update the model using $D_j = d_j^*$ and $S_{j+1} = s_{j+1}$;

remove arcs: A_{j+1}--->D_{j+2},…, D_{n-1}

add the arc: S_{j+1}--->D_{j+2}

end if

recover decision nodes in day $j+2, j+3, \dots, n$ to be uniform.

updated model:

$$\Psi_A = \Psi_A(d_1^*, \dots, d_j^*, \dots, D_{n-1}, \dots, a_2, \dots, a_{j+1}, \dots, A_n, V)$$

$$\Psi_D = \Psi_D(d_1^*, \dots, d_j^*, \dots, D_{n-1}, \dots, a_2, \dots, a_{j+1}, \dots, A_n, V)$$

end for

output:

Updated model:

$$\Psi_A = \Psi_A(d_1^*, \dots, d_{n-1}^*, a_2, \dots, a_n, s_2, \dots, s_n, V)$$

$$\Psi_D = \Psi_D(d_1^*, \dots, d_{n-1}^*, a_2, \dots, a_n, s_2, \dots, s_n, V)$$

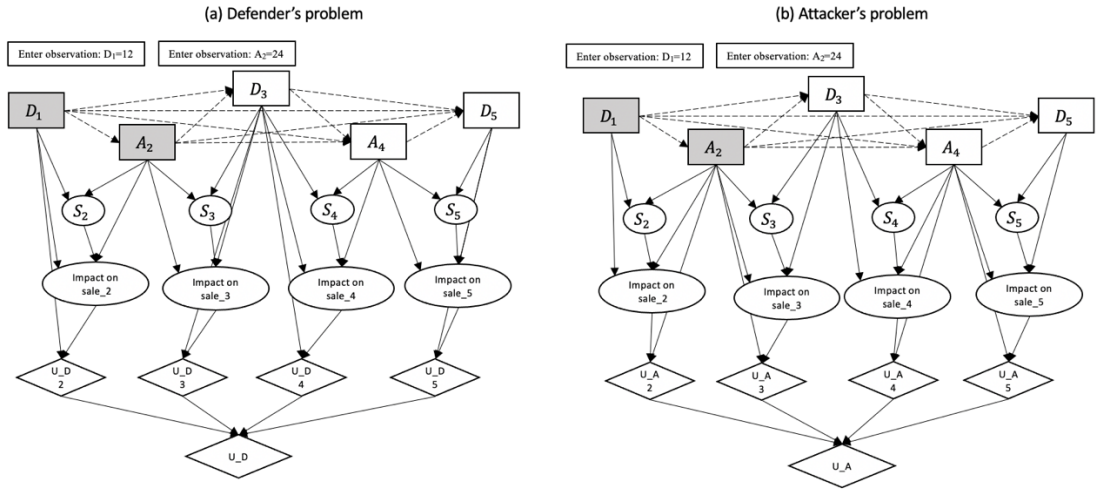
Optimal strategies for the defender:

$$\{d_1^*, \dots, d_{n-1}^*\}$$

7.2.2 Example 3: the Actual Attacks are Observable

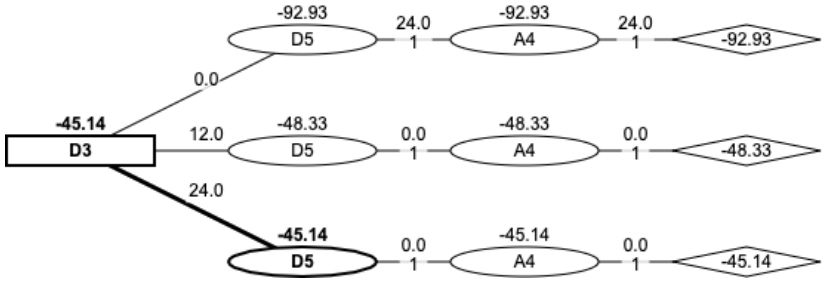
In this subsection, we show how we can apply algorithm 2 to support dynamic decision-making based on example 2 with the actual attacks are observable. According to the results calculated in Example 2, $\{d_1^* = 12, a_2^* = 0, d_3^* = 12, a_4^* = 0, d_5^* = 0\}$, the defender would deploy $D_1 = d_1^* = 12$ on Monday as her optimal move. However, when it comes to Wednesday and the defender is going to deploy the defence, we assume she realizes that the attacker attacks i.e., 24 hours on Tuesday rather than 0 hour represented by $a_2^* = 0$. The strategy for this is to use $D_1 = d_1^* = 12$ and $A_2 = a_2 = 24$ updating the model. We enter observations of D_1 and A_2 into the model shown in Figure 7-3. The updated HBN representing the defender's problem and the attacker's problem on Wednesday is shown in Figure 7-9.

Figure 7-9 The updated HBNs for the DDM on Wednesday.



Then to calculate the optimal strategy for the defender standing on Wednesday, we apply Algorithm 2 to the updated HBNs. The optimal strategy for the defender on Wednesday is represented by the DT in Figure 7-10. Hence, we get $d_3^* = 24$.

Figure 7-10 The DT of D3 given information observed before Wednesday



Decision trees constructed in the process are shown in Figure 7-11 and Figure 7-12.

Figure 7-11 The DT of D5 given information observed on Wednesday

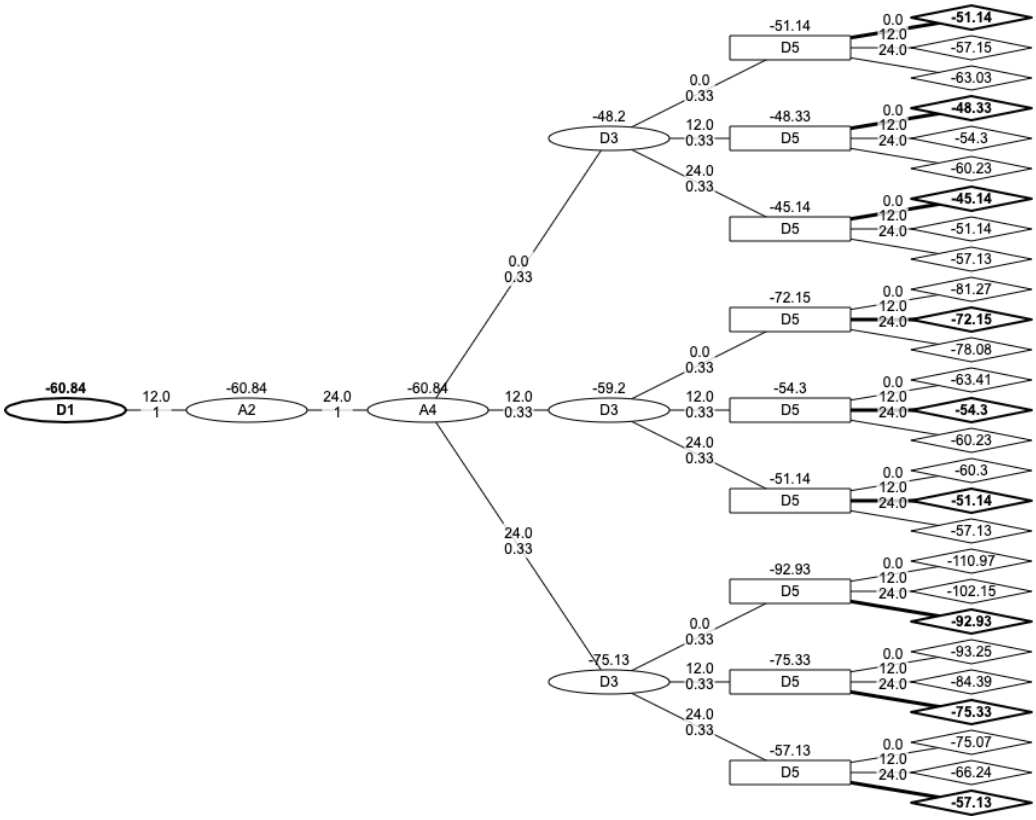
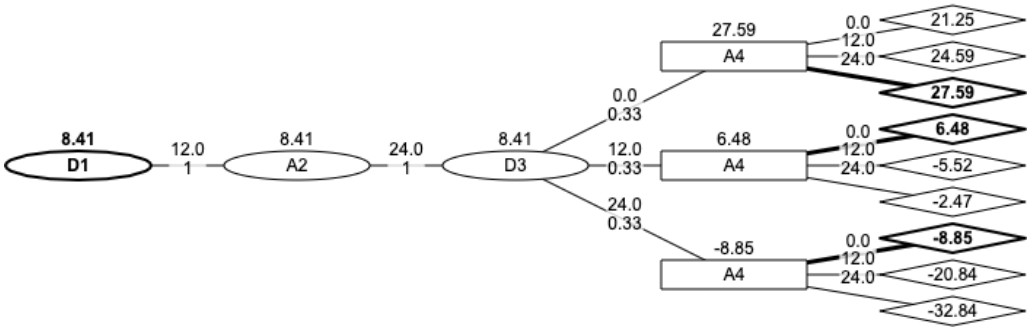
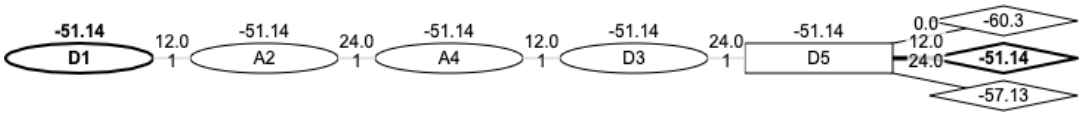


Figure 7-12 The DT of A4 given information observed on Wednesday



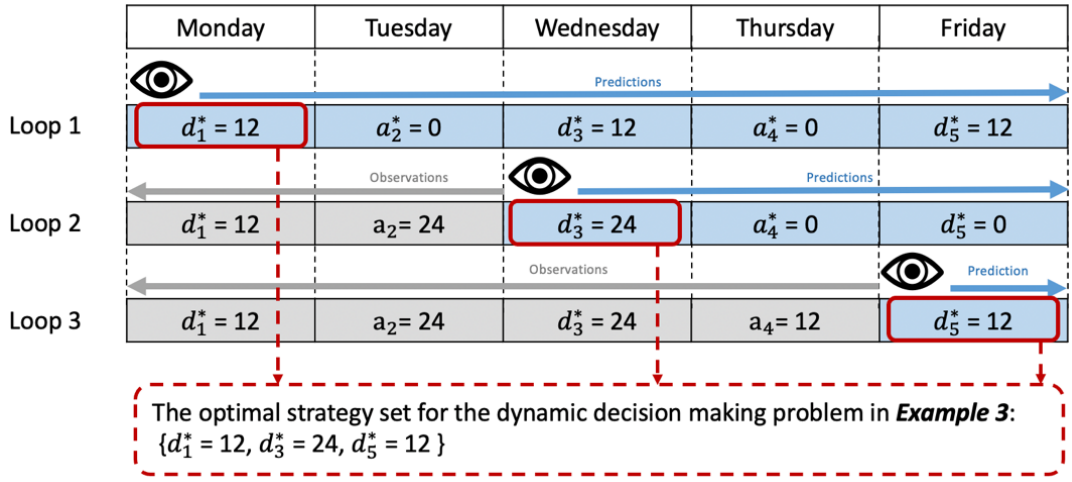
Finally, when it comes to Friday, given observed attack on Thursday, we can calculate the defender's optimal choice following the similar way. Assuming that the attacker eventually adopted $A_4 = a_4 = 12$ on Thursday, we update the model by entering the observations $D_3 = d_3^* = 24$ and $A_4 = a_4 = 12$. Then we construct a DT of D_5 to determine the best choice for the defender when she makes the defence decision on Friday. We show this DT in Figure 7-13, which represents that $D_5 = d_5^* = 12$ is the optimal choice when the defender stands on Friday.

Figure 7-13 The DT of D5 given information observed on Friday



Hence, we support the defender with optimal decisions in the dynamic process that deploy D_1 = 12 at the first place, deploy D_3 = 24 when observing the attack on Tuesday is A_2 = 24 and deploy D_5 = 12 when observing the attack on Thursday is A_4 =12. We illustrate observations, predictions and the optimal strategy set of this dynamic decision-making process in Figure 7-14.

Figure 7-14 Results summary of dynamic decision making in Example 3

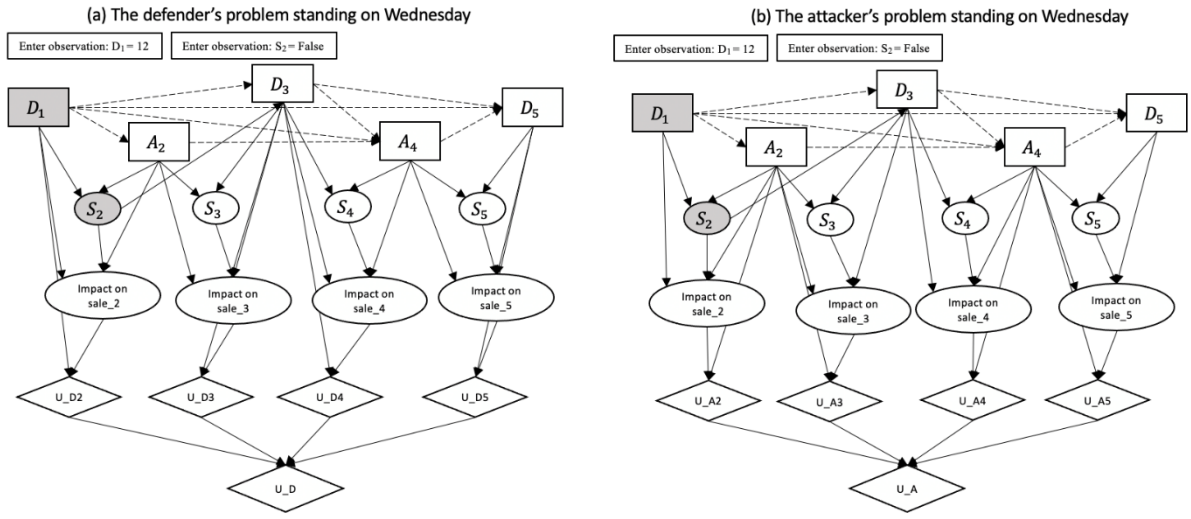


7.2.3 Example 4: only the Consequences of Attacks are Observable

In this subsection, we illustrate how we apply algorithm 2 to support dynamic decision-making based on example 2 with only the consequences of attacks are observable.

Let's start from considering how to determine the optimal decision for the defender when it comes to Wednesday. At this time point, the defender has conducted $D_1 = d_1^* = 12$ on Monday and can observe if the attack succeeded on Tuesday (represented by the node S_2). We remove the arc pointing from node A_2 to D_3, since under the assumption of this example, the past attack is no more observable. In addition, to represent D_3 is influenced by S_2, we add an arc pointing from S_2 to D_3. Then we update the model using the real-time data which includes $D_1 = d_1^* = 12$ and the observed states of S_2 (i.e., assuming S_2 = False). The updated model is shown in Figure 7-15. Algorithm 2 can be then applied to the updated HBNs, where D_1 and S_2 are chance nodes while D_3, A_4 and D_5 are decision nodes. The corresponding optimal strategy set can be then calculated.

Figure 7-15 The updated HBNs for the DDM problem on Wednesday.



By constructing the DT for the defender on Wednesday, which is represented in Figure 7-16, we know that the optimal decision for the defender standing on the current time point (Wednesday) is $d_3^* = 12$. The optimal strategy set for all the decision nodes at this stage is $\{d_3^* = 12, a_4^* = 0, d_5^* = 0\}$. DTs constructed in the process are shown in Figure 7-17 and Figure 7-18.

Figure 7-16 The DT of D3 given information observed on Wednesday

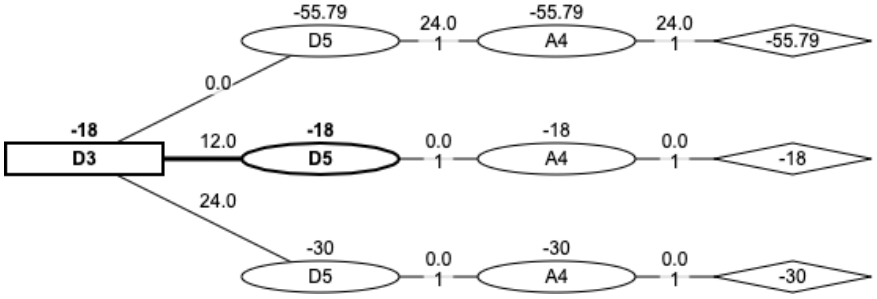


Figure 7-17 The DT of D5 given information observed on Wednesday

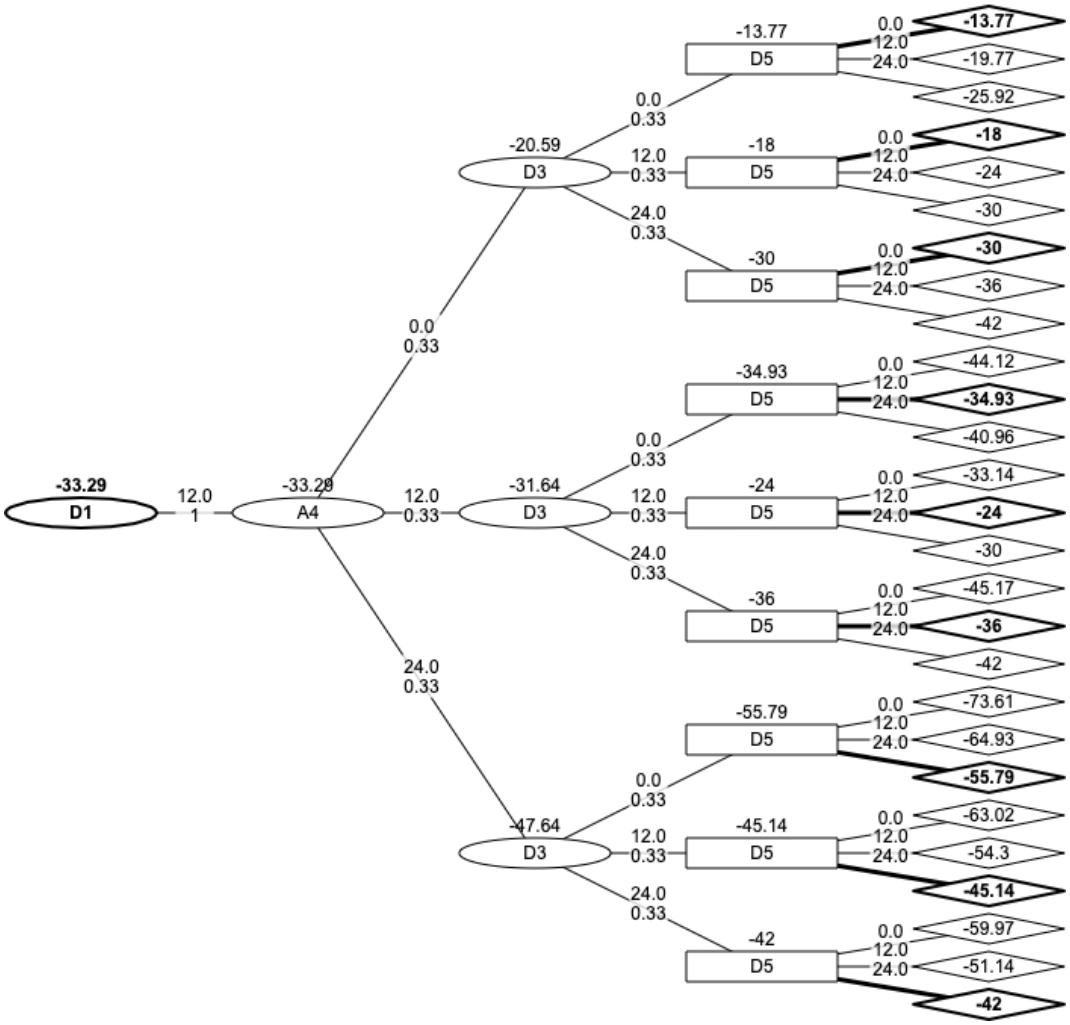
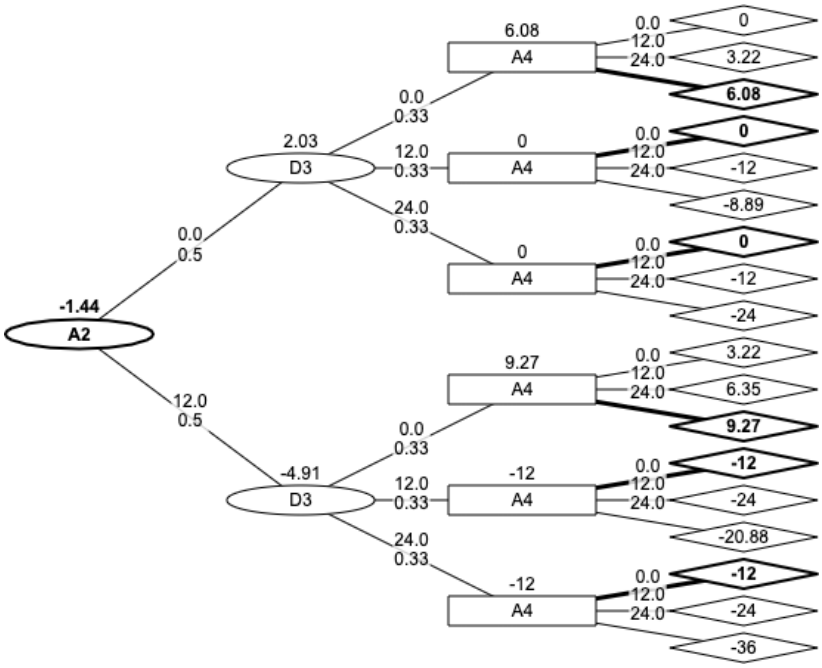


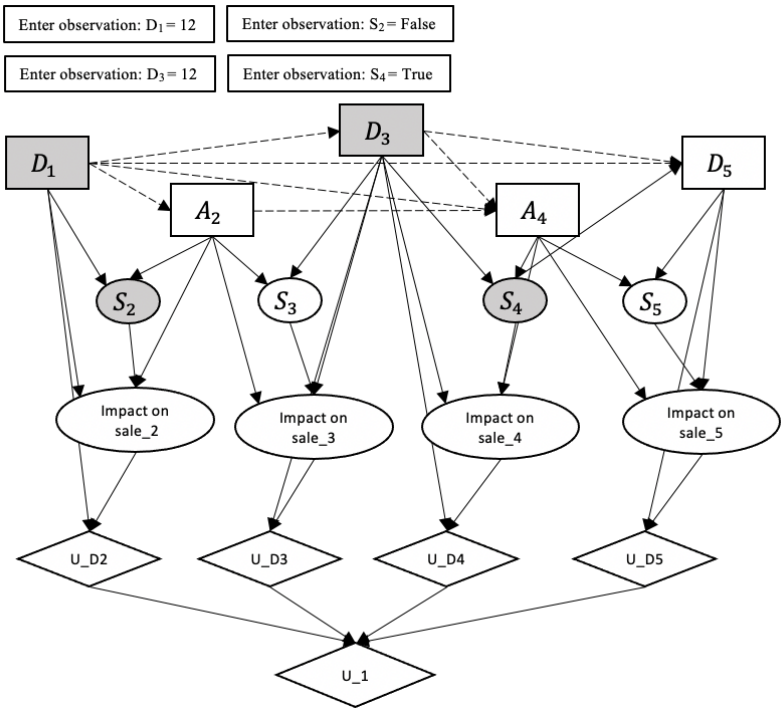
Figure 7-18 The DT of A4 given information observed on Wednesday



The newly calculated strategy d_3^* is the optimal one for the defender at the current time point (Wednesday) given observed information have been considered to update her mind while she still uses her best knowledge to predict what attack would be adopted on Thursday.

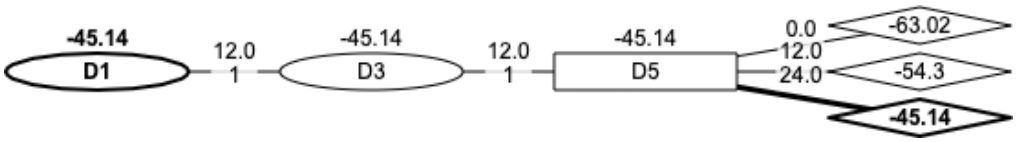
The defender would then adopt this optimal strategy on Wednesday. When it comes to Friday, she needs to update her mind again with the newly observed information and furthermore determines the best move on Friday. We can assume she observes that the attack on Thursday succeeded. Since the attack on Thursday is unobservable in reality, we remove the arc pointing from node A_4 to D_5. Meanwhile, to represent D_5 is influenced by S_4, we add an arc pointing from S_4 to D_5. In this stage, the decision node is only D_5 and there are no attack decisions needed to be predicted for determining the optimal D_5. The updated HBN is shown in Figure 7-19.

Figure 7-19 The updated HBN for the defender's problem on Friday.



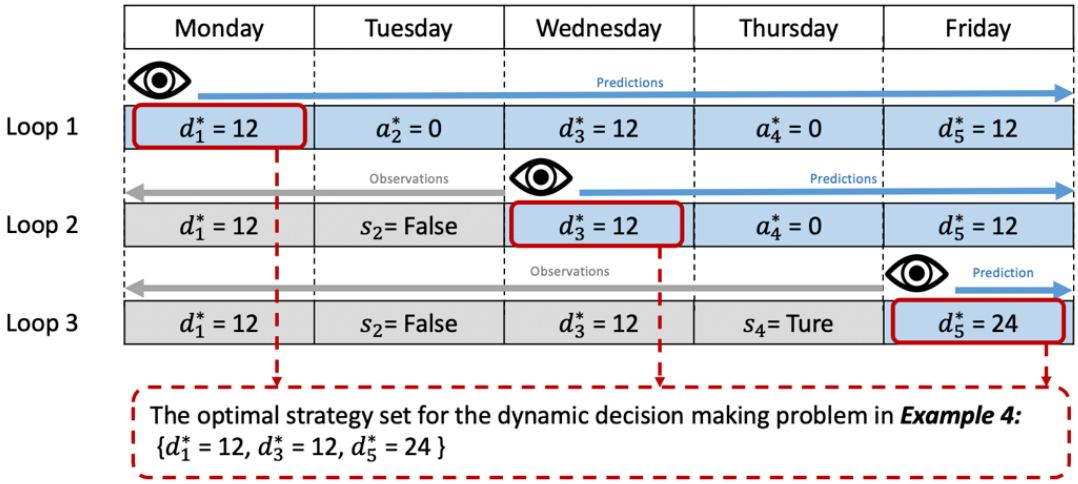
Based on the updated model, we construct the corresponding DT which is shown in Figure 7-20. It can be represented that the optimal strategy for the defender standing on Friday is $d_5^* = 24$.

Figure 7-20 The DT of D5 given information observed on Friday



Hence, we support the defender with optimal decisions in the dynamic process that deploy $D_1 = 12$ at the first place, deploy $D_3 = 12$ when observing the attack on Tuesday fails and deploy $D_5 = 12$ when observing the attack on Thursday successes. We illustrate observations, predictions and the optimal strategy set of this dynamic decision-making process in Figure 7-21.

Figure 7-21 Results summary of dynamic decision making in Example 4



7.3 Summary

We propose an HBN based ARA approach for supporting decision making in sequential defend-attack game problems. This kind of problem is typically extracted to the sequential D-A model. We illustrate how to use the proposed method to calculate the optimal strategy in this template. Furthermore, to model more complicated cases that may be likely in practice, we consider two extended sequential D-A templates involving extra variables and longer decision sequence respectively. We construct the algorithm based on HBNs and the ARA approach to calculate optimal decisions for the supported agent (the defender) and provide examples to illustrate how the proposed method can be applied. Since the applied HBN inference provides an automated way to compute hybrid D-A models and extends their use to involve mixtures of continuous and discrete variables, the proposed HBN based ARA approach is more versatile compared with the Monte Carlo (MC) based ARA approach. More importantly, the proposed approach is novel in that it supports dynamic decision making whereby new real-time observations can be employed to update the D-A model timely and optimal decisions can be determined based on both generic information from the past and rigorous anticipation about the future. This dynamic decision analysis mechanism can make more effective use of information and can better simulate the actual decision-making process. Examples are provided, illustrating how the proposed

framework can be adjusted for decision analysis in more complicated but more practical scenarios and serving as template for further expansion according to practical application requirement.

Chapter 8 Risk Analysis and Decision Supporting Using the HBN Framework

In this chapter, we use examples to illustrate how the CRA models built using HBNs can be incorporated for customized CRA requirement. This is achieved by incorporating the FAIR-BN with a process-oriented model and a game-theoretic model to provide integrated risk assessment. We call the extend FAIR-BNs as EFBNs which in tandem illustrate the expandability of FAIR-BNs.

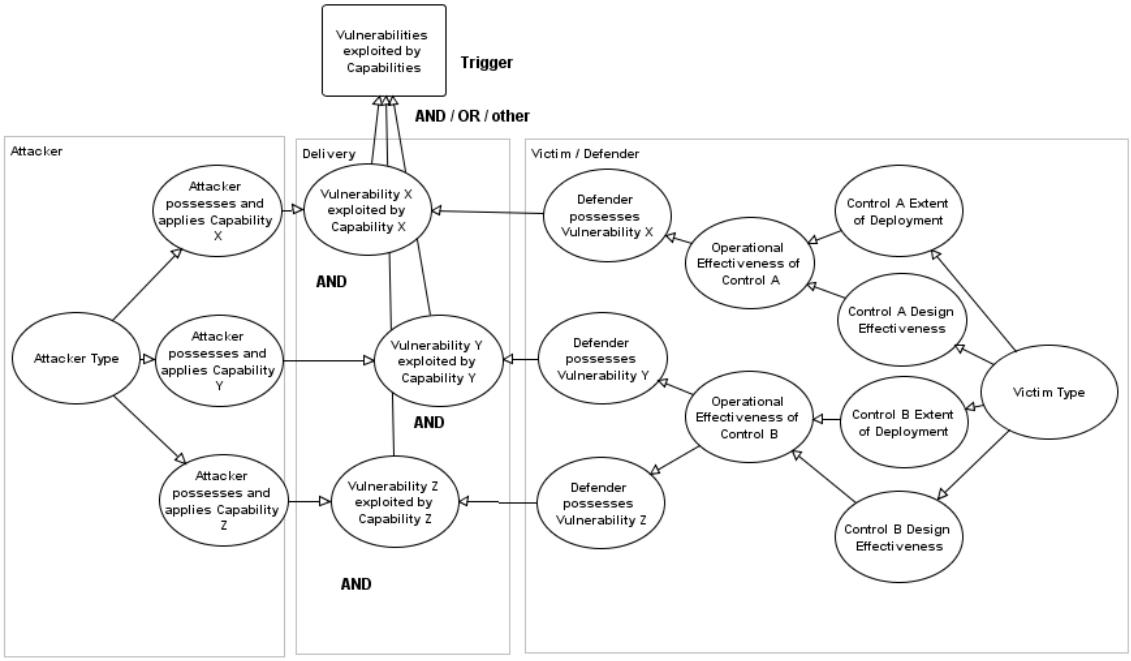
8.1 Extending the FAIR-BN Using a Process-Oriented Model

In addition to providing more flexibility when modelling input distributions and providing more accurate results, as we illustrated in Chapter 5, 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, a control deployment model, into the FAIR-BN, replacing the calculation of the vulnerability variable in FAIR by a richer causal structure.

We construct the control deployment model as shown in Figure 8-1. 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

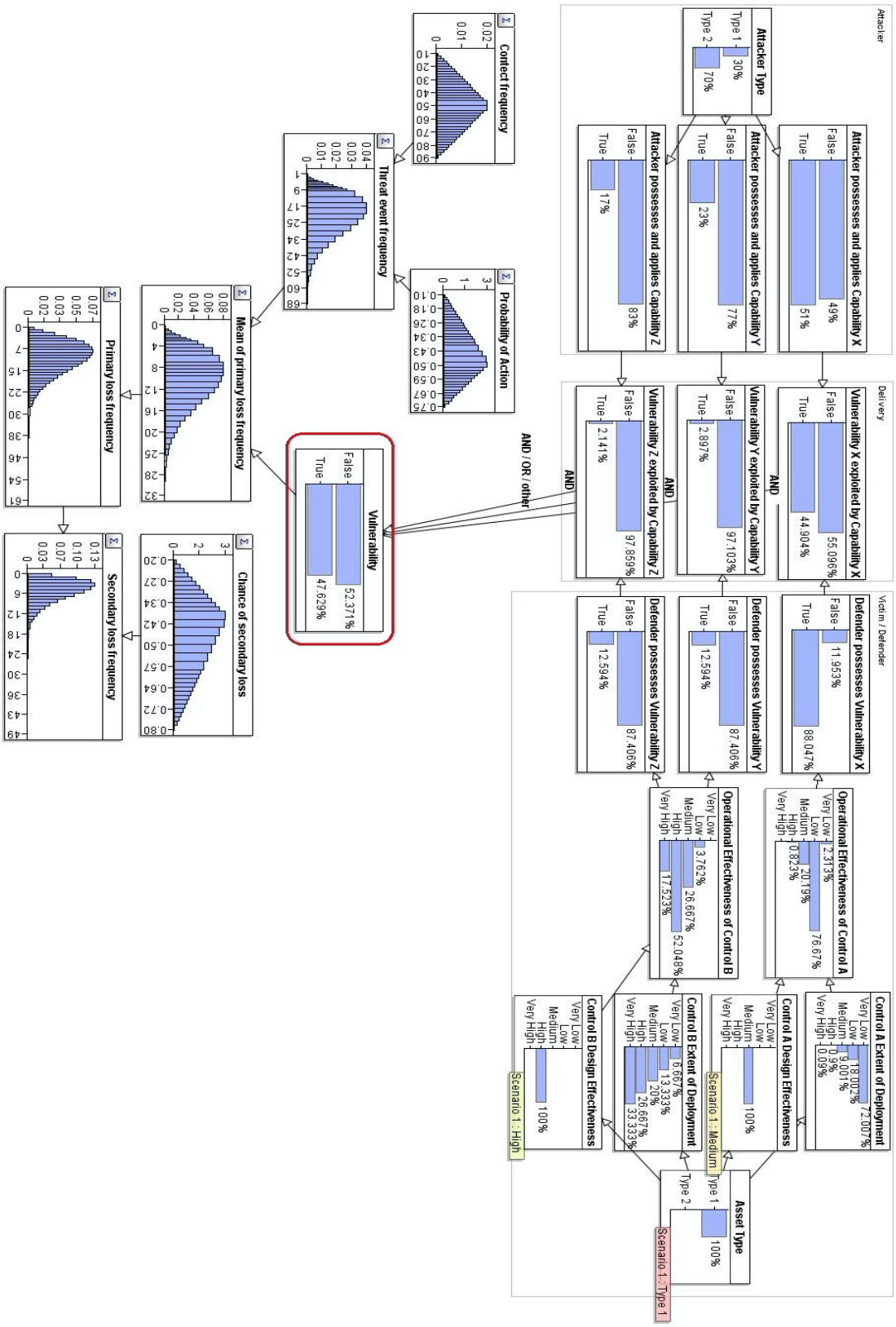
using simple Boolean “AND” and “OR” relationships and CPTs could be elicited from expert knowledge.

Figure 8-1 A control deployment scenario



The probabilities used in this model are an example, which will not influence the reasoning mechanism which we have described in chapter 3. The model is illustrated in Figure 8-2. Similarly, other process-oriented risk assessment models, such as the kill chain model [42] and attack graphs [45, 46] can be combined with the FAIR-BN for more advanced risk assessment.

Figure 8-2 FAIR-BN extended by a process-oriented model

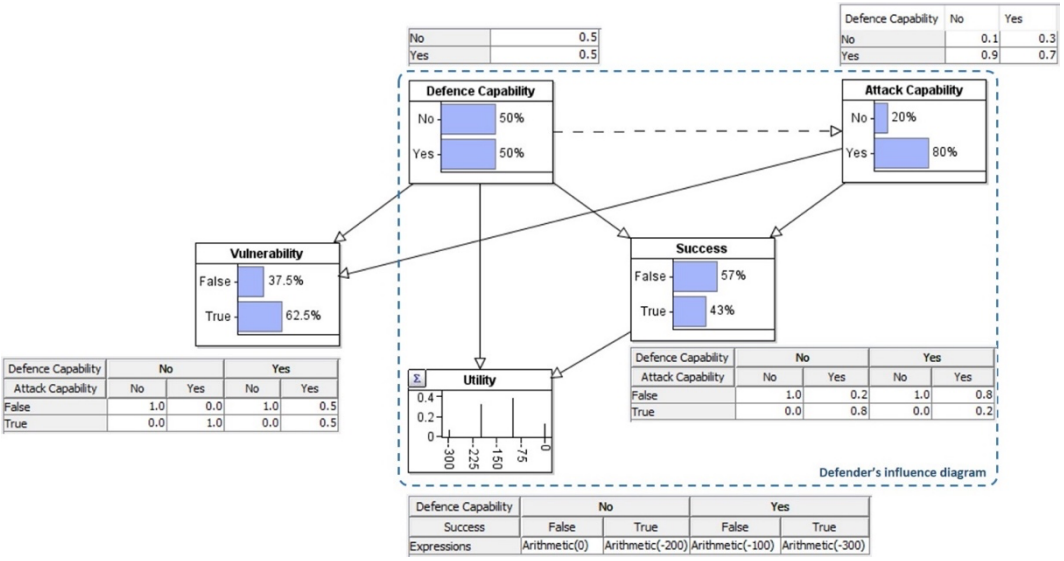


8.2 Connecting FAIR-BN with Adversarial Risk Analysis

In the FAIR model, the vulnerability of an information asset is determined by a contest between attackers and the defender. Here we show how a sequential defend-attack game model can be accommodated to enhance the FAIR model by considering the game between the defender and the attacker and in tandem construct an EFBN.

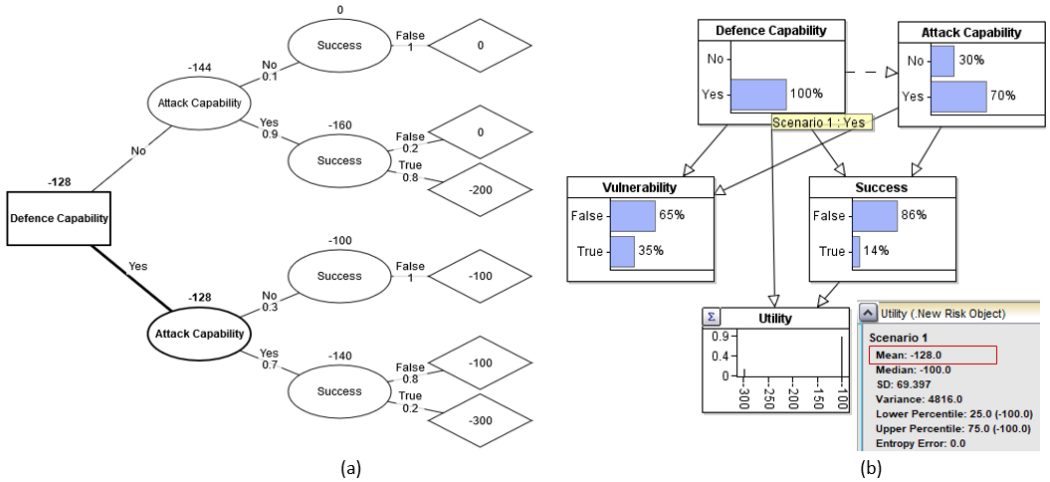
An example using adversarial risk analysis is shown in Figure 8-3, which is represented by an Influence Diagram (ID) built with a BN. We assume that the defender's decision is about whether to equip the capability of a defence, d, to protect a target information asset. Meanwhile, after observing the defender's capability, the attacker would consider whether to deploy a corresponding attack 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 Figure 8-3. 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.

Figure 8-3 The BN according to the defender's ID



Typically, the aim in decision analysis is to maximize the utility node of the supported decision maker. Corresponding to the ID shown in Figure 8-3, a Decision Tree (DT) can be generated using AgenaRisk. We show this DT in Figure 8-4 (a). The applied algorithms and details for generating DTs from hybrid IDs in AgenaRisk are described in [130]. 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 Figure 8-4 (b), which can be then used in our FAIR-BN for further analysis.

Figure 8-4 Decision results of the defender's ID



(a) (b)

8.3 Summary

Based on the empirical evaluation of FAIR and the analyse of its the rigidity, we show 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) by incorporating FAIR-BN with a process-oriented model and a defend-attack game model, which can be further expanded for more complicated scenarios as we tackle with in Chapter 7. EFBNs 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.

Chapter 9 Conclusion

This thesis has explored the application of Bayesian network technology in cyber risk problems focusing on addressing three challenges in cybersecurity risk modelling as described in Section 1.1. Specifically, seven objectives of this thesis are addressed correspondingly mapped to each of these challenges. In this section, we show how these objectives are achieved in this thesis.

*For **modelling complex cyber problems with explanatory risk estimation,** our objective is set and achieved as below:*

***Objective I:** Introduce causal analysis and related Bayesian network technology to underpin cybersecurity risk modelling and provide an example using the kill chain model.*

This objective is achieved in Chapter 2 and Chapter 3. We reviewed CRA standards and approaches in Chapter 2 and claim it is possible to avoid the limitations of current standards and approaches by using causal and probabilistic analysis. We provide the explanation in section 2.2 and introduce the characteristics of causal risk analysis in section 2.3. Since causal risk analysis is implemented using Bayesian networks, we provide related knowledge in Chapter 3. In section 3.1, the background knowledge of Bayesian networks is provided, then in section 3.2, we show how to construct Bayesian networks from expert knowledge, using idioms, which can be used as building blocks for efficient and high-quality modelling and specify the CPTs of involved variables utilizing both historical data and probabilities which represent personal belief. In section 3.3, we introduce the core algorithms, which are the dynamic discretisation algorithm and the junction tree algorithm, that guarantee efficient inference in HBNs. HBNs have been extended, in the form of Influence Diagrams (IDs), to support decision making tasks. We have introduced this in section 3.4. In additional, we provide examples using BNs to model concrete cybersecurity problem in section 3.5. Specifically, a kill chain model is implemented using Bayesian networks to illustrate how causal analysis can be applied in cybersecurity risk modelling.

*For **conducting CRA from an organizational perspective**, our objective is set and achieved as below:*

***Objective II:** Reveal the limitations of the FAIR model.*

This objective is achieved in Chapter 4. In this chapter, we provide an introduction of Factor Analysis of Information Risk (FAIR) and revealing a number of important limitations that FAIR has, based on an in-depth analysis of the FAIR model's assumptions, focusing on its taxonomic structure and algorithms. This analysis has hitherto not appeared in the literature. We find that the FAIR algorithms restrict both the type of statistical distributions that can be used and the expandability of the model structure. Moreover, since the FAIR model uses only triangular distributions to simulate input risk factors of the model, alternative statistical distributions (especially long-tailed distributions) for input factors may be poorly approximated, and thus inaccuracy is introduced. The related analysis and claims can be supported by Publication 1.

***Objective III:** Create a BN alternative to the FAIR model which eliminates FAIR's restrictions and delivers improved practical utility.*

This objective is achieved in Chapter 5 and Chapter 8. In section 5.1, we illustrate how we construct the BN alternative to the FAIR model which can adopt the same modelling assumptions used by the FAIR model. We call this alternative as FAIR-BNs. The proposed FAIR-BNs allow a wider set of distributions to represent and process input variables compared with the FAIR model. This is also illustrated by an example in section 5.1.3. In addition, the FAIR-BN can be easily extended by incorporating with other CRA models built using BNs to enhance the analysis. 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 in Chapter 8. This is covered by Publication 1.

***Objective IV:** Evaluate the accuracy of the FAIR and FAIR-BN approaches and identify the pros and cons in both approaches.*

This objective is achieved in Chapter 5. We propose a quantitative accuracy evaluation method using results generated by the FAIR-MC and the measurement *J* divergence in Section 5.2 and conduct experimental test of FAIR and FAIR-BN in Section 5.3 and Section 5.4. 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. This can be supported by Publication 1.

*For **conducting CRA from a technical perspective,** our objective is set and achieved as:*

***Objective V:** Propose an HBN-based alternative framework for ARA and identify its advantages compared with the state of the art.*

This objective is achieved in Chapter 6. In this chapter, we provide the introduction of ARA and illustrate the calculation mechanism of ARA focusing on a typical game model, sequential Defend-Attack (D-A) models, for it can properly represent realistic cybersecurity cases. We propose an alternative framework, based on HBN inference and decision trees, to solve the typical sequential D-A games from the ARA perspective, which is represented in Section 6.3. In Section 6.4, we illustrate how to conduct the calculation and support the decision making. We have pointed out the advantages of the proposed framework compared with other previous works in Section 6.1. Most ARA solutions use Monte Carlo (MC) simulation to carry out the calculation. The pro of the MC simulation is that it is straightforward to implement, while the cons include 1) it can become computationally challenging when dealing with complex decision dependent uncertainties; 2) it cannot dynamically cope with new evidence that could be used to update the game model in real time, which is a realistic requirement for practical use. In comparison, the

main advantages of the proposed framework are: 1) it offers a fully automated way to compute hybrid D-A models which involve mixtures of continuous and discrete variables; 2) it supports dynamic decision making in multi-period D-A games which has not been solved by previous ARA solutions that adopt MC simulation. The later advantage is illustrated in chapter 7. This can be supported by Publication 2.

Objective VI: *Apply the proposed framework to solve more practical D-A problems, i.e., involving extra risk variables and longer decision sequence.*

This objective is achieved in Chapter 7 especially in Section 7.1. In this section, we summarize the rules needed to be followed to extend D-A models for more complicated scenarios, i.e., D-A problems with extra variables and with longer decision sequences. Two examples are provided to illustrate how these rules are applied and work out. This can be supported by Publication 3.

Objective VII: *Provide a mechanism in the framework to support dynamic decision making in multi-period D-A games and present working examples.*

This objective is achieved in Chapter 7 especially in Section 7.2. In this section, we discuss dynamic decision analysis in multi-period D-A games with the algorithm is provided in section 7.2.1 and examples are provided in section 7.2.2 and 7.2.3. The proposed approach supports dynamic decision making whereby new real-time observed information can be employed to update the D-A model timely and optimal decisions can be determined based on both generic information from the past and rigorous anticipation about the future. This dynamic decision analysis mechanism can make more effective use of information and can better simulate the actual decision-making process for fulfilling practical application requirements. This part of work is covered by Publication 3.

We have proposed an CRA framework which improves the related approaches from several aspects that are stated corresponding to the seven objectives. Here we summarize the pros and cons of our framework compared with the related approaches from three perspectives. Firstly, in terms of the general cybersecurity risk analysis, the idea of standard impact-based risk measurement, which is adopted

by many of the standards and approaches discussed in section 2.2, can guide a convenient preliminary CRA in practice. This kind of risk measurement, which usually evaluates risk based on the scale of its possibility and impact, is much easier to understand and implement without high-technical requirements nor time-consuming calculation process. When there are higher requirements on a unified understanding of risk (i.e., in cross-group cooperation), explanatory modelling of risk, and sufficient details for risk evaluation and the related decision making, causal risk analysis can be a better choice. Secondly, when we analyze cybersecurity risk from an organizational perspective, we propose FAIR-BN and FAIR-MC based on the FAIR model. 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. In analysing cybersecurity from a technical perspective, the game between the defender and the attacker, we have proposed an HBN-based ARA approach based on the MC based ARA approach. Both approaches can be the ideal choice for static decision analysis. More importantly, the HBN-based ARA approach is novel in that it supports dynamic decision making whereby new real-time observations are used to ensure that the calculated optimal decision is based on the maximized use of existing information. To the best of our knowledge, this has not been achieved by the current MC based ARA approach.

How the proposed framework can be applied in actual cases can be vary from the purpose, the level of requirement and from the different groups and organizations. Our framework has provided a CRA foundation that can tackle with the risk modelling and calculation problems and can be easily customized by the Cybersecurity Risk Management (CRM) practitioners for practical application. We analyze the application of the framework from two aspects which are the CRA process it focuses on and how it can fit into the entire CRM picture.

As being introduced in section 2.1, CRA consists of three phases in general, which are risk identification, estimation, and evaluation. The proposed framework has provided the means for these three phases from both organizational and technical perspectives. From the organizational perspective, the preliminary practice of CRA could be: 1) (risk identification) identify risk factors of information assets using the FAIR structure; 2) (risk estimation) calculate the organization's potential financial losses related to each information asset being attacked using the FAIR-BN; 3) (risk evaluation, which is also a decision making process here) prioritize information assets to be protected according to their associated financial losses and the organization's criteria, i.e., the security budget of the organization. From the technical perspective, the HBN based ARA approach has provided the means to organize risk modelling and calculation and provided the decision analysis mechanism which can assist the defender to find the optimal defence strategy and get the maximum utility. The example provided by subsection 7.1.2 has illustrated how the HBN based ARA approach can be applied in practice for conducting CRA. It is because, in this example, the data and related assumptions are derived from an actual case provided by [125]. We have also illustrated how the CRA from these two perspectives can be enriched by incorporating with each other or other causal models (in Chapter 8). Hence, we believe our framework has provided a CRA foundation for practical usage and can be easily customized by CRM practitioners for specific actual cases.

The whole CRM process is an iterative process of reviewing and monitoring risks. It can be represented by Figure 1-1 which is provided by the international CRM standard ISO/IEC 27005 [6]. This standard has provided detailed guidance for conducting CRM and has been widely recognized and applied in industry. Our proposed framework can also be used in the context of ISO/IEC 27005 standard. It can be a supplement to ISO/IEC 27005 (and other CRA standard/framework), for enhancing the risk assessment as can be illustrated by Figure 4-1 [97]. Therefore, the proposed framework can be applied by organizations which has adopted ISO/IEC 27005 as their guidance for conducting CRM. This also guarantees that the proposed framework can be applied in industrial practice.

In summary, the proposed Bayesian network based CRA framework can be used as a foundation or a guidance for causal/probabilistic cybersecurity risk modelling and calculating in general and specifically in risk assessment from the organizational level and the technical level. By adopting Bayesian network technology, the framework can effectively organize the risk modelling referring causal idioms, employ expert's knowledge and empirical data while embrace uncertainty, update the risk assessment model using real-time observed information and provide mechanism for dynamic decision analysis. This framework also has flexibility to be adjusted and expanded with other BN based models for fulfilling customized requirements of CRA in practice, and therefore provides a means to narrow the gap between research and industrial implements. To achieve more potential benefits of this framework, some further research incorporating the framework with practical requirements could be undertaken. In the proposed framework, the critiria for risk evaluation is based on financial losses the organization may encounter if certain information asset/system been attacked. Richer critiria for risk evaluation can be considered based on the proposed framework, i.e., taking insurance into consideration. The work introduced in [125] can be an example for this.

Bibliography

1. Hubbard, D.W. and R. Seiersen, *How to measure anything in Cybersecurity risk*. 2016: John Wiley & Sons.
2. GOV.UK, *NATIONAL CYBER SECURITY STRATEGY 2016-2021*. 2016, https://www.gov.uk/government/publications/national-cyber-security-strategy-2016-to-2021.
3. *ISO/IEC 27001: 2013 Information technology, security techniques, information security management systems, requirements*. 2013.
4. Blakley, B., E. McDermott, and D. Geer. *Information security is information risk management*. in *Proceedings of the 2001 workshop on New security paradigms*. 2001. ACM.
5. Young, C., *Metrics and methods for security risk management*. 2010: Syngress.
6. 27005, I.I., *ISO/IEC 27005: 2011 Information technology, security techniques, information security risk management*. 2011
7. Peltier, T.R., *Information security risk analysis*. 2010: Auerbach publications.
8. *ISO/IEC 27002: 2013 Information technology, security techniques, code of practice for information security management*. 2013.
9. Shameli-Sendi, A., R. Aghababaei-Barzegar, and M. Cheriet, *Taxonomy of information security risk assessment (ISRA).* Computers & security, 2016. **57**: p. 14-30.
10. Wangen, G., C. Hallstensen, and E. Snekkenes, *A framework for estimating information security risk assessment method completeness.* International Journal of Information Security, 2016: p. 1-19.
11. Sendi, A.S., et al. *FEMRA: Fuzzy expert model for risk assessment*. in *2010 Fifth International Conference on Internet Monitoring and Protection*. 2010. IEEE.
12. Manshaei, M.H., et al., *Game theory meets network security and privacy.* ACM Computing Surveys (CSUR), 2013. **45**(3): p. 25.
13. Do, C.T., et al., *Game theory for cyber security and privacy.* ACM Computing Surveys (CSUR), 2017. **50**(2): p. 30.
14. Roy, S., et al. *A survey of game theory as applied to network security*. in *2010 43rd Hawaii International Conference on System Sciences*. 2010. IEEE.

15. Wang, Y., et al. *A survey of game theoretic methods for cyber security*. in *2016 IEEE First International Conference on Data Science in Cyberspace (DSC)*. 2016. IEEE.
16. Landoll, D.J. and D. Landoll, *The security risk assessment handbook: A complete guide for performing security risk assessments*. 2005: CRC Press.
17. Wheeler, E., *Security risk management: Building an information security risk management program from the Ground Up*. 2011: Elsevier.
18. NVD, *National vulnerabilities database.* < http://nvd.nist.gov/cpe.cfm >
19. Stango, A., N.R. Prasad, and D.M. Kyriazanos. *A threat analysis methodology for security evaluation and enhancement planning*. in *2009 Third International Conference on Emerging Security Information, Systems and Technologies*. 2009. IEEE.
20. Farahmand, F., et al. *Managing vulnerabilities of information systems to security incidents*. in *Proceedings of the 5th international conference on Electronic commerce*. 2003.
21. NIST, *SP 800-30 Rev.1, Guide for Conducting Risk Assessments*. 2012.
22. Hulitt, E. and R.B. Vaughn, *Information system security compliance to FISMA standard: a quantitative measure.* Telecommunication Systems, 2010. **45**(2): p. 139-152.
23. Farahmand, F., et al., *A management perspective on risk of security threats to information systems.* Information Technology and Management, 2005. **6**(2): p. 203-225.
24. Sun, L., R.P. Srivastava, and T.J. Mock, *An information systems security risk assessment model under the Dempster-Shafer theory of belief functions.* Journal of Management Information Systems, 2006. **22**(4): p. 109-142.
25. Shameli-Sendi, A., M. Cheriet, and A. Hamou-Lhadj, *Taxonomy of intrusion risk assessment and response system.* Computers & Security, 2014. **45**: p. 1-16.
26. Jones, A., *A framework for the management of information security risks.* BT technology journal, 2007. **25**(1): p. 30-36.
27. Cherdantseva, Y., et al., *A review of cyber security risk assessment methods for SCADA systems.* Computers & security, 2016. **56**: p. 1-27.

28. Jones, J., *An introduction to factor analysis of information risk (fair).* Norwich Journal of Information Assurance, 2006. **2**(1): p. 67.
29. Freund, J. and J. Jones, *Measuring and managing information risk: a FAIR approach*. 2014: Butterworth-Heinemann.
30. Yazar, Z., *A qualitative risk analysis and management tool–CRAMM.* SANS InfoSec Reading Room White Paper, 2002. **11**: p. 12-32.
31. Stolen, K., et al. *Model-based risk assessment–the CORAS approach*. in *NIK (2002) informatics conference, Kongsberg*. 2002.
32. ISACA *The Risk IT Framework*. 2009.
33. Caralli, R.A., et al., *Introducing octave allegro: Improving the information security risk assessment process*. 2007, CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.
34. Ekelhart, A., et al. *Security ontologies: Improving quantitative risk analysis*. in *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)*. 2007. IEEE.
35. Asosheh, A., B. Dehmoubed, and A. Khani. *A new quantitative approach for information security risk assessment*. in *2009 2nd IEEE International Conference on Computer Science and Information Technology*. 2009. IEEE.
36. Cobit, S., *A business framework for the governance and management of enterprise IT.* Rolling Meadows, 2012.
37. Fenton, N. and M. Neil, *Risk assessment and decision analysis with Bayesian networks Second Edition*. 2019: Crc Press.
38. LeBlanc, D. and M. Howard, *Writing secure code*. 2002: Pearson Education.
39. Schneier, B., *Attack trees.* Dr. Dobb's journal, 1999. **24**(12): p. 21-29.
40. Jha, S., O. Sheyner, and J. Wing. *Two formal analyses of attack graphs*. in *Computer Security Foundations Workshop, 2002. Proceedings. 15th IEEE*. 2002. IEEE.
41. Bistarelli, S., F. Fioravanti, and P. Peretti. *Defense trees for economic evaluation of security investments*. in *Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on*. 2006. IEEE.
42. Hutchins, E.M., M.J. Cloppert, and R.M. Amin, *Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill*

chains. Leading Issues in Information Warfare & Security Research, 2011. **1**(1): p. 80.

43. Khand, P.A. *System level security modeling using attack trees*. in *Computer, Control and Communication, 2009. IC4 2009. 2nd International Conference on*. 2009. IEEE.
44. Byres, E.J., M. Franz, and D. Miller. *The use of attack trees in assessing vulnerabilities in SCADA systems*. in *Proceedings of the international infrastructure survivability workshop*. 2004.
45. Poolsappasit, N., R. Dewri, and I. Ray, *Dynamic security risk management using bayesian attack graphs.* IEEE Transactions on Dependable and Secure Computing, 2012. **9**(1): p. 61-74.
46. Liu, Y. and H. Man. *Network vulnerability assessment using Bayesian networks*. in *Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security 2005*. 2005. International Society for Optics and Photonics.
47. Xie, P., et al. *Using Bayesian networks for cyber security analysis*. in *Dependable Systems and Networks (DSN), 2010 IEEE/IFIP international conference on*. 2010. IEEE.
48. Pearl, J., *Fusion, propagation, and structuring in belief networks.* Artificial intelligence, 1986. **29**(3): p. 241-288.
49. Pearl, J., *Probabilistic reasoning in intelligent systems: networks of plausible inference*. 1988: Morgan Kaufmann Publishers Inc. 552.
50. Marquez, D., M. Neil, and N. Fenton, *A new Bayesian Network approach to Reliability modelling*, in *5th International Mathematical Methods in Reliability Conference*. 2007: Glasgow.
51. Fenton, N. and M. Neil, *Risk assessment and decision analysis with Bayesian networks*. 2012: Crc Press.
52. Marsh, W., et al. *Using operational data for decision making: a feasibility study in rail maintenance*. in *Safety and Reliability*. 2016. Taylor & Francis.
53. Heckerman, D., A. Mamdani, and M.P. Wellman, *Real-world applications of Bayesian networks.* Communications of the ACM, 1995. **38**(3): p. 24-26.
54. Chockalingam, S., et al. *Bayesian network models in cyber security: a systematic review*. in *Nordic Conference on Secure IT Systems*. 2017. Springer.

55. Wang, J.A. and M. Guo. *Vulnerability categorization using Bayesian networks*. in *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research*. 2010. ACM.

56. Pecchia, A., et al. *Identifying compromised users in shared computing infrastructures: A data-driven bayesian network approach*. in *Reliable Distributed Systems (SRDS), 2011 30th IEEE Symposium on*. 2011. IEEE.

57. Axelrad, E.T., et al. *A Bayesian network model for predicting insider threats*. in *Security and Privacy Workshops (SPW), 2013 IEEE*. 2013. IEEE.

58. Apukhtin, V., *Bayesian network modeling for analysis of data breach in a bank*. 2011, University of Stavanger, Norway.

59. Mo, S.Y.K., P.A. Beling, and K.G. Crowther. *Quantitative assessment of cyber security risk using bayesian network-based model*. in *2009 Systems and Information Engineering Design Symposium*. 2009.

60. Mihajlovic, V. and M. Petkovic, *Dynamic bayesian networks: A state of the art.* 2001.

61. Frigault, M., et al. *Measuring network security using dynamic bayesian network*. in *Proceedings of the 4th ACM workshop on Quality of protection*. 2008. ACM.

62. Liu, Y. and H. Man. *Network vulnerability assessment using Bayesian networks*. in *Proc. SPIE*. 2005.

63. Pan, S., et al. *Causal event graphs cyber-physical system intrusion detection system*. in *Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop*. 2013. ACM.

64. Yet, B., et al., *An improved method for solving hybrid influence diagrams.* International Journal of Approximate Reasoning, 2018. **95**: p. 93-112.

65. Khosravi-Farmad, M., et al. *Network security risk mitigation using Bayesian decision networks*. in *Computer and Knowledge Engineering (ICCKE), 2014 4th International eConference on*. 2014. IEEE.

66. Koller, D. and B. Milch, *Multi-agent influence diagrams for representing and solving games.* Games and economic behavior, 2003. **45**(1): p. 181-221.

67. Lin, P., M. Neil, and N. Fenton, *Risk aggregation in the presence of discrete causally connected random variables.* Annals of Actuarial Science, 2014. **8**(2): p. 298-319.

68. *Agena Ltd. 2002-2019. AgenaRiskV10 software package, www.AgenaRisk.com.*

69. Fenton, N. and M. Neil, *The use of Bayes and causal modelling in decision making, uncertainty and risk.* CEPIS Upgrade, 2011. **12**(5): p. 10-21.

70. Zhou, Y., N. Fenton, and M. Neil, *Bayesian network approach to multinomial parameter learning using data and expert judgments.* International Journal of Approximate Reasoning, 2014. **55**(5): p. 1252-1268.

71. Zhou, Y., et al. *Incorporating expert judgement into Bayesian network machine learning*. in *Twenty-Third International Joint Conference on Artificial Intelligence*. 2013.

72. Neil, M., N. Fenton, and L. Nielson, *Building large-scale Bayesian networks.* The Knowledge Engineering Review, 2000. **15**(3): p. 257-284.

73. Fenton, N., M. Neil, and D.A. Lagnado, *A general structure for legal arguments about evidence using Bayesian networks.* Cognitive science, 2013. **37**(1): p. 61-102.

74. Kuipers, B., A.J. Moskowitz, and J.P. Kassirer, *Critical decisions under uncertainty: Representation and structure.* Cognitive Science, 1988. **12**(2): p. 177-210.

75. Gigerenzer, G. and U. Hoffrage, *How to improve Bayesian reasoning without instruction: Frequency formats.* Psychological Review, 1995. **102**(4): p. 684-704.

76. Diez, F.J., *Parameter adjustment in Bayes networks. the generalized noisy OR-gate*, in *Proceedings of the Ninth international conference on Uncertainty in artificial intelligence*. 1993, Morgan Kaufmann Publishers Inc.: Washihgton, DC. p. 99-105.

77. Akaike, H., *A Bayesian analysis of the minimum AIC procedure.* Annals of the Institute of Statistical mathematics, 1978. **30**(1): p. 9-14.

78. Cooper, G.F., *A simple constraint-based algorithm for efficiently mining observational databases for causal relationships.* Data Mining and Knowledge Discovery, 1997. **1**(2): p. 203-224.

79. Cheng, J., D.A. Bell, and W. Liu. *Learning belief networks from data: An information theory based approach*. in *Proceedings of the sixth international conference on Information and knowledge management*. 1997. ACM.

80. Cooper, G.F. and E. Herskovits, *A Bayesian method for the induction of probabilistic networks from data.* Machine learning, 1992. **9**(4): p. 309-347.

81. Heckerman, D., D. Geiger, and D.M. Chickering, *Learning Bayesian networks: The combination of knowledge and statistical data.* Machine learning, 1995. **20**(3): p. 197-243.

82. Lam, W. and F. Bacchus, *Learning Bayesian belief networks: An approach based on the MDL principle.* Computational intelligence, 1994. **10**(3): p. 269-293.

83. Larrañaga, P., et al., *Structure learning of Bayesian networks by genetic algorithms: A performance analysis of control parameters.* IEEE transactions on pattern analysis and machine intelligence, 1996. **18**(9): p. 912-926.

84. Lauritzen, S.L. and F. Jensen, *Stable local computation with conditional Gaussian distributions.* Statistics and Computing, 2001. **11**(2): p. 191-203.

85. Murphy, K.P. and S. Russell, *Dynamic bayesian networks: representation, inference and learning.* 2002.

86. Spiegelhalter, D., et al., *BUGS 0.5: Bayesian inference using Gibbs sampling manual (version ii).* MRC Biostatistics Unit, Institute of Public Health, Cambridge, UK, 1996: p. 1-59.

87. BUGS, *BUGS (Bayesian inference Using Gibbs Sampling)*. 1989: Available at: https://www.mrc-bsu.cam.ac.uk/software/bugs (Accessed: Aug, 2017).

88. Kozlov, A.V. and D. Koller. *Nonuniform dynamic discretization in hybrid networks*. in *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*. 1997. Morgan Kaufmann Publishers Inc.

89. Cooper, G.F., *The computational complexity of probabilistic inference using Bayesian belief networks.* Artificial intelligence, 1990. **42**(2-3): p. 393-405.

90. Lauritzen, S.L. and D.J. Spiegelhalter, *Local computations with probabilities on graphical structures and their application to expert systems.* Journal of the Royal Statistical Society: Series B (Methodological), 1988. **50**(2): p. 157-194.

91. Agena, *AgenaRisk*. 2016: Available at: http://www.agenarisk.com (Accessed: Aug, 2017).

92. Sendi, A.S. and M. Cheriet. *Cloud computing: a risk assessment model*. in *2014 IEEE International Conference on Cloud Engineering*. 2014. IEEE.

93. Le, A., et al., *Assessing loss event frequencies of smart grid cyber threats: Encoding flexibility into fair using bayesian network approach*, in *Smart Grid Inspired Future Technologies*. 2017, Springer. p. 43-51.

94. Le, A., et al., *Incorporating FAIR into Bayesian Network for Numerical Assessment of Loss Event Frequencies of Smart Grid Cyber Threats.* Mobile Networks and Applications, 2018: p. 1-9.

95. Park, M., et al., *Situational Awareness Framework for Threat Intelligence Measurement of Android Malware.* JoWUA, 2018. **9**(3): p. 25-38.

96. *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.

97. Group, T.O., *FAIR – ISO/IEC 27005 Cookbook* 2010.

98. *OPEN FAIR™ TOOL WITH SIPMATH™ DISTRIBUTIONS: GUIDE TO THE THEORY OF OPERATION*. 2018.

99. *THE OPEN FAIR™ RISK ANALYSIS TOOL*

100. *The Open FAIR™ tool with Sipmath™ Distributions: Guide to the theory of operation. https://publications.opengroup.org/g181*. 2018, The Open Group.

101. *The Open FAIR™ risk analysis tool. https://publications.opengroup.org/i181*. 2019, The Open Group.

102. Anderson, C. and M.P. Andersson, *Long tail.* 2004.

103. Foss, S., D. Korshunov, and S. Zachary, *Heavy-tailed and long-tailed distributions*, in *An Introduction to Heavy-Tailed and Subexponential Distributions*. 2013, Springer. p. 7-42.

104. Tankard, C., *Advanced persistent threats and how to monitor and deter them.* Network security, 2011. **2011**(8): p. 16-19.

105. Banks, D.L., J.M.R. Aliaga, and D.R. Insua, *Adversarial risk analysis*. 2015: Chapman and Hall/CRC.

106. Heckman, P.E. and G.G. Meyers. *The calculation of aggregate loss distributions from claim severity and claim count distributions*. in *Proceedings of the Casualty Actuarial Society*. 1983.

107. *THE OPEN FAIR™ RISK ANALYSIS TOOL*.

108. Keelin, T.W., *The metalog distributions.* Decision Analysis, 2016. **13**(4): p. 243-277.

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

110. Lin, J., *Divergence measures based on the Shannon entropy.* IEEE Transactions on Information theory, 1991. **37**(1): p. 145-151.

111. Robert, C.P., *Simulation of truncated normal variables.* Statistics and computing, 1995. **5**(2): p. 121-125.

112. Mahadevan, S., *Monte carlo simulation.* MECHANICAL ENGINEERING-NEW YORK AND BASEL-MARCEL DEKKER-, 1997: p. 123-146.

113. Kullback, S. and R.A. Leibler, *On information and sufficiency.* The annals of mathematical statistics, 1951. **22**(1): p. 79-86.

114. *MATLAB 2018a, The MathWorks, Inc., Natick, Massachusetts, United States.*

115. Danielsson, P.-E., *Euclidean distance mapping.* Computer Graphics and image processing, 1980. **14**(3): p. 227-248.

116. *THE OPEN FAIR™ RISK ANALYSIS TOOL.* https://publications.opengroup.org/i181.

117. Joshi, C., J. Rios, and D.R. Insua, *Insider threat modeling: An adversarial risk analysis approach.* IEEE Transactions on Information Forensics and Security, 2020.

118. Gindis, H., *The Bounds of Reason: Game theory and the Unification of the Behavioural Sciences*. 2009, Princeton Univ. Press.

119. Rios Insua, D., J. Rios, and D. Banks, *Adversarial risk analysis.* Journal of the American Statistical Association, 2009. **104**(486): p. 841-854.

120. Brown, G., et al., *Defending critical infrastructure.* Interfaces, 2006. **36**(6): p. 530-544.

121. Zhuang, J. and V.M. Bier, *Balancing terrorism and natural disasters—Defensive strategy with endogenous attacker effort.* Operations Research, 2007. **55**(5): p. 976-991.

122. Hausken, K. and V.M. Bier, *Defending against multiple different attackers.* European Journal of Operational Research, 2011. **211**(2): p. 370-384.

123. Rios Insua, D., D. Banks, and J. Rios, *Modeling opponents in adversarial risk analysis.* Risk Analysis, 2016. **36**(4): p. 742-755.

124. González-Ortega, J., D.R. Insua, and J. Cano, *Adversarial risk analysis for bi-agent influence diagrams: An algorithmic approach.* European Journal of Operational Research, 2019. **273**(3): p. 1085-1096.

125. Insua, D.R., et al., *An adversarial risk analysis framework for cybersecurity.* arXiv preprint arXiv:1903.07727, 2019.

126. Gil, C. and J. Parra-Arnau, *An Adversarial-Risk-Analysis Approach to Counterterrorist Online Surveillance.* Sensors, 2019. **19**(3): p. 480.

127. Aliprantis, C.D. and S.K. Chakrabarti, *Games and decision making.* OUP Catalogue, 2012.

128. *Agena Ltd. 2002-2021*, w.A.c. AgenaRiskV10 software package, Editor., AgenaRiskV10 software package, www.AgenaRisk.com.: AgenaRiskV10 software package, www.AgenaRisk.com.

129. Ekin, T., et al., *Augmented Probability Simulation Methods for Non-cooperative Games.* arXiv preprint arXiv:1910.04574, 2019.

130. Fenton, N. and M. Neil, *Risk assessment and decision analysis with Bayesian networks Second Edition*. 2018: Crc Press.