Context- and Bias-Free Probabilistic Mission Impact Assessment

Alexander Motzek and Ralf Möller
Universität zu Lübeck
Email: motzek@ifis.uni-luebeck.de, moeller@uni-luebeck.de

Technical Report

Abstract

Assessing and understanding the impact of scattered and widespread events onto a mission is a pertinacious problem. Current approaches attempting to solve mission impact assessment employ score-based algorithms leading to spurious results. We identify a fourfold problem with score-based algorithms: (1) score-based algorithms enforce deep training of experts to employed frameworks for specification (non-context-free), (2) require reference results for interpreting obtained results (non-bias-free), (3) require assessments outside of an experts' expertise (non-local), and (4) require validation of end-results against ground truth. This paper provides a formal, mathematical model for bias- and context-free mission impact assessment. Based on a probabilistic model we reduce mission impact assessment to a well-understood mathematical problem based on definitions from local expertise and allow for a validation at data level. This is useful for areas and applications where qualitative assessments are required, such as assessments in critical infrastructures or military contexts.

1 Introduction

Modeling dependencies of missions on various involved resources is a novel field of research, which pursues the goal of assessing the influences of local impacts on some resources onto a higher goal, i.e., a mission. Assessments somehow require an approach to "spread" locally created impacts onto higher goals, such as missions or processes. Early approaches attempting to solve a problem of mission impact assessment use ad-hoc methods involving newly established algorithms. We argue that such newly created algorithms suffer from multiple discrepancies, which we categorize into four different groups: (1) An expert must first fully understand and be trained in a system before he can assess configurations and parameters. We say, such systems do not provide a context-free assessment. (2) Obtained results from a system require a steep learning curve for interpretation and easily lead to overfitting by a dulling due to learned reference values. This means, results are not bias-free and require knowledge about a system. (3) During configurations, experts are forced outside their expertise, leading to potentially inaccurate specifications. We argue that it is favorable to accept disagreement from multiple knowledge sources instead of enforcing the definition of one allegedly congruent knowledge base. Finally, configurations (compare Problem 1 and 3) were assessed by a possibly overtrained expert and might be inaccurate, but parameters are not verifiable nor can be validated by an independent third party. This means, (4) obtained results from a newly created algorithm must be validated against a ground truth. Ground truths for occurred events and their exact impact on a mission are often not available in large quantities or are confidential.

To put things into perspective, a non-context-free system requires an expert to understand how an evaluation reacts to a parameter of "5" and how to a parameter of "3"—without the context of the complete framework, such values do not have any meaning and are neither verifiable, validatable, nor are understandable. Further, an end-user becomes biased from interpretation of received results: With an unclear, non-mathematical definition of an end-result, e.g., "yellow," "3," or "severe," an end-user intrinsically adapts over time to "normal" results and becomes biased, i.e., a reported "severe" "red" error of category "5" is first taken serious, if it persists for an hour.

In this work, we take a view from different perspectives towards mission impact assessment. We consider three views from three experts from different expertise and combine them inside a well-defined probabilistic graphical model. We provide a context-free assessment of defined parameters and models, which are assessable and can be validated by themselves without knowing their later use. Based on this probabilistic model one finds a well-understood problem: In a complex network multiple events possibly occur, whose local effects must be assessed towards a global effect. Using a probabilistic approach, one can benefit from existing, well-defined and well-understood algorithms to solve this problem and obtain probabilistically sound results that are understandable without the knowledge of our approach. Obtained results are understandable using commonsense and do not suffer from biased interpretations. Furthermore, we present results of two real world use cases on real data using our approach.

The contribution of this article can be summarized as follows: By introducing a well-defined probabilistic graphical model for mission impact assessment, we are able to reduce impact assessment on a well-defined mathematical problem, which allows for a validation of results at data level and does not require deep training of experts. By resorting to local conditional probability distributions one is able to integrate widespread knowledge from different expertise into one sound model. This is useful for applications, where qualitative assessments are required and perpendicular views from multiple experts onto a problem must be brought inline. As a long term goal this provides the basis for an automated response system based on a mathematical well-defined model for risk and impact assessments in a predictive and retrospect analysis over time in changing and dynamic environments.

Scope and structure of this article The remainder of this article is structured as follows: In Section 2 we discuss related work and identify common discrepancies and benefits of existing approaches. Based on a well-defined probabilistic graphical model, we develop in Section 3 a mathematical model for mission impact modeling based on views from different experts. We introduce a notion of temporal aspects to cover dynamic environments to a certain degree and propose an independent-timeslice model assessing impacts at independent points in time, e.g., at independent short-, mid- and long-term time points. Based on this model, we discuss mission impact assessment as a formalized problem and its theoretical complexity in Section 4, and evaluate the scalability and accuracy of a proposed algorithm in largely scaled domains. In Section 5 we apply the introduced independent-timeslice mission impact assessment in two real world use cases involving business-, IT-, and security experts from different domains and show that the approach delivers satisfying and greatly accepted results.

Based on recent advantages in artificial intelligence and probabilistic graphical models, we dedicate Section 6 to an outlook of future work on an extension of the presented independent-timeslice model towards completely dynamic probabilistic mission impact assessments for rapidly changing environments and time-dependent analyses at a, if intended, nearly continuous time granularity. We discuss and propose various approaches for such an extension and show how derived independent-timeslice models can be reused directly in future work. We conclude in Section 7.

2 Related Work

Mission modeling and mission impact assessment is an emerging field of research. While existing approaches deliver early results and claim to solve mission impact assessment, a formal definition of an underlying problem is yet missing, which leads to the mentioned problems of biased interpretation of results and a non-context free parametrization. Employed fudge factors in newly established algorithms lead to untraceable and spurious results demanding data driven validations. Unfortunately, large, standardized datasets for validation are yet missing for mission impact assessment and in the following presented work.

Barreto et al. [1] introduce a well-understood modeling technique and use BPMN models to acquire knowledge. An impact assessment is based on various indexes and numerical scores, such as exploit index, impact factor, infrastructure capacity index, and graph distances. Various numerical factors are arbitrarily combined, without a mathematical foundation and cannot provide a transparent, understandable and verifiable assessment to an expert. Further, an assessment is solely based on direct impacts, leaving aside transitive impacts and/or defining a manual description of all dependencies between individual devices inside one organization, which is, in most of the cases an unfeasible process. Albanese et al. present in [2] a well-modeled formalism for complex inter-dependencies of missions as a set of tasks. Using numerical scores and tolerances in a holistic approach Albanese et al. focus on cost minimization. Their approach can solely be validated holistically, as involved parameters do not bear local semantics and do not provide bias-free and context-free understandable results. Buckshaw et al. [3] propose a quantitative risk management by involving various experts and present a score-based assessment based on individual values and a standardization using a weighted sum. Unfortunately, a mathematical foundation is missing and obtained results are only interpretable after deep training of experts in the characteristics of this approach. Buckshaw et al. themselves note that a validation of the proposed model requires large amounts of actual data and ground truth, which both are not available.

Jacobson [4] presents a well understood conceptual framework using interdependencies based on operational capacity at different abstraction layers. In this dependency model, impacts are propagated and reduce the operational capacity, which has a similar intention to our approach. However, [4] uses self-defined metrics for propagating impacts through Boolean gates, which can not provide context- and bias-free understandable results or parametrization. Moreover, an explicit representation of "intra-asset" dependencies is required, i.e., all individual critical, and non-critical resources must be identified. [5] proposes the use of BPMN models and describes a process for evaluating impacts of cyber attacks. However, [5] fails to get across any mathematical approaches or formal definitions for impact assessment.

Further works focuses solely on modeling. For example, Goodall et al. [6] focus on modeling and available data integration using ontologies but do not address an impact assessment. Another ontology-based approach is presented by D'Amico et al. in [7] and identifies multiple experts while noting that, e.g., system administrators are not capable of understanding an organization's missions.

In terms of (probabilistic) approaches towards assessments of impacts caused by vulnerabilities and attacks, probabilistic models have been researched by Wang et al. [8], Liu et al. [9] or Xie et al. [10]. However, Wang et al. base their work on attack graphs and do not consider imperfect knowledge, e.g., unknown extents of damage causable by vulnerabilities, uncertainity of specific events and potentially disagreeing information sources as we do. Xiep et al. and Liu et al. are significantly limited by the lack of supporting cyclic dependencies and do not consider any mission impact relations. Other impact propagation approaches, e.g., by Kheir et al. [11] or Jahnke et al. [12], claiming to handle such details, are not probabilistic based and degrade to a handcrafted propagation algorithm with arbitrary scores, where parameters are only assessable by a trained expert and obtained results can only be used in a holistic way, as they provide no directly interpretable meaning. By the reduction of mission impact assessments to probabilistic inference in probabilistic graphical models, our approach provides a context-free parametrization that does not require deep training of experts, merges multiple sources of information, and delivers results which are directly interpretable without requiring reference results and are suitable for qualitative situation reports.

3 Dependencies and Impacts

In the following, we take a view from different perspectives towards mission impact assessment. We consider three views from three experts from different expertise. We do not enforce an expert to overlook assessments from other experts and expertises (local assessment), and further, do not require that an expert understands or is trained on how his assessment are used inside algorithms and frameworks (context-free assessment). Based on a probabilistic model, we are able to include all three views directly into one consistent model bridging semantic and technology gaps. Based on this model a well-understood probabilistic inference problems is evident that assesses a mission impact from widespread events towards a bias-free and context-free understandable result.

Every expert defines a different dependency model, where every modeled entity represents a random variable and a dependency between two entities is represented by a local conditional probability.

Remark 1 (Impact)**.** *We use an abstract term of "impact" in our work in the sense of "not operating as fully intended." The underlying meaning of "intended operation" lies in the use case of a model.* ▲

3.1 Mission Dependency Model (Business View)

In order to perform a mission impact assessment, one must understand the good one aims to protect, e.g., a mission or a company. In the field of business intelligence, a complete company or organization, i.e., the good one aims to protect, is modeled as a conglomeration of *business processes*. Commonly, business processes are modeled using the business process modeling notation (BPMN) and a business process is modeled as a (dependent) collection of tasks. This modeling approach is well accepted and can be found, e.g., in [1, 2, 5]. Figure 1 shows a sketch of a BPMN model used throughout this paper.

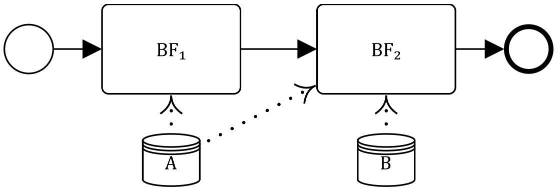

Figure 1: Example BPMN 2.0 model sketch for the BP_1 business process shown in the dependency model of Figure 2.

Designing BPMN models is handled manually by an expert from a company or by an external business consultant having a precise expertise in the understanding of business analysis. The business analysis is performed on a pure business perspective and stops at a "resource" level, e.g., it identifies a web-service, but does not describe the dependencies of the webservice on a database or a data center. This is a reasonable approach, as the latter perspective comes from a very different expertise and would require very broad-range experts. Further, an identification of a web-service as a business relevant object is causally more precise in the terms of an operation perspective: A failing database might cause a web-service to not operate as intended and therefore might lead to an unaccomplishment of a business-process. Still, the direct cause for the business-process being unaccomplishable is a rogue web-service and *not* the database. In contrary, an "IT" expert might identify a web-service to be irrelevant, as the crucial point of failure or the point of interest lies in the availability of data from a database. Nevertheless, the latter dependencies must be covered and are discussed in the upcoming subsection.

We extend a model by Jakobson [4] and model mission dependencies as shown in Figure 2 as a graph of *mission nodes*. We model a *company* as being dependent on its *business processes*. A business process is again dependent on one or more *business functions*. *Business resources* provide business functions. Business resources are part of an infrastructure perspective and—from an operational view—might be irrelevant, but were identified to be business critical. Figure 2 shows a dependency graph of business relevant objects, based on the preceding presented BPMN model.

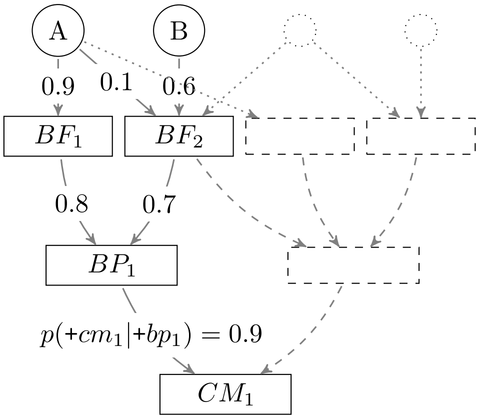

Figure 2: Mission Dependency Model. Values along edges denote individual conditional probability fragments. Solid entities correspond to Figure 1 and are used in Example 1. Consequences and further attributes are omitted in this figure.

Comparing Figure 2 to the original BPMN model in Figure 1, relations between BPMN entities and mission nodes become evident: Each BPMN model represents one business process inside a company, where BPMN tasks represent business function. Likewise, BPMN tasks access data from data stores, representing immediately critical business resources. Later, in Section 5 we give a real world example for generating and obtaining a mission dependency model in collaboration with business and IT experts.

We model every dependency, i.e., edge in Figure 2, as a local conditional probability. Every conditional probability describes a probability of failure if a dependency fails. For example, the probability of the business-function BF_1 (see Figure 2 and 1), e.g., "provide access to customer data," failing, given the required business-resource A, e.g., "customer-data-frontend," fails is 90%. We argue that the meaning of local conditional probabilities are understandable using common-sense (e.g., "in 9 out of 10 cases, customer data was not accessible for employees during frontend-server maintenance") and that the (numerical) assessment can be directly validated by either an expert or through ground-truth.

Definition 1 (Probabilistic Preliminaries)**.** *Every node inside a dependency model represents a random variable, denoted as capital X, where every random variable is assignable to one of its possible values $x \in \mathrm{dom}(X)$. Let $P(X = x)$ denote the probability of random variable X having x as a value. For our case we consider* $\mathrm{dom}(X) = \{true, false\}$ *and we write $+x$ for the event $X = true$ and $\neg x$ for $X = false$.* ▲

The event $+x$ represents the case that node X is operationally *impacted* and $\neg x$ that is operating as fully intended, i.e., no impact is present. Given a random variable X (e.g., a business process) that is dependent on multiple random variables $\vec{Z}$ (e.g., a set of business functions), the dependencies of X on all $Z \in \vec{Z}$ are represented by one conditional probability distribution (CPD) $P(X|\vec{Z})$ of node X. For ease of parametrization of these distributions we use commonly known combination functions, such as noisy-or and noisy-and. By the use of combination functions, not complete conditional probability distributions must be parametrized, but solely single fragments of a distribution for every dependency, or edge, in a model.

Definition 2 (From Dependencies to Distributions)**.** *We render every dependency of random variable Y on X as an* individual *conditional probability $p(+x|+y)$ and $p(+x|\neg y)$. Such individual conditional probability are* fragments *of a complete conditional probability distribution (CPD) and are therefore denoted in lowercase. To acquire the local CPD $P(X|\vec{Y})$ of node X from* all *its individual dependencies on nodes $Y \in \vec{Y}$, we employ noisy-and and noisy-or combination functions, as, e.g., described by Henrion in [13].* ▲

For the scope of this work we consider non-leaky combination functions. Non-leakiness implies that a source for an impact must origin from inside a model and cannot occur "from nowhere," i.e., $P(+x|\neg\vec{y}) = 0$, therefore $p(+x|\neg y) = 0$ is fixed for every dependency. We believe that for most mission nodes a suitable combination function is noisy-or, representing that every impact on a dependence might lead to an impact on the dependent node, i.e., every impacted dependence is a sufficient cause for an impact. If dependencies are laid out completely redundantly, a noisy-and combination function can be used for mission nodes to directly reflect redundancies. Notwithstanding, if an expert feels confident to do so, CPDs can be designed directly. Formally, we define a mission dependency model as follows.

Definition 3 (Mission Dependency Model)**.** *A mission dependency model M is a directed acyclic graph (DAG) as a pair $\langle\vec{V},\vec{E}\rangle$ of vertices $\vec{V}$ and edges $\vec{E}$. Vertices $\vec{V}$ are random variables (Def. 1) and are categorized according to their semantic as business-resources ($\vec{BR}$), -functions ($\vec{BF}$), -processes ($\vec{BP}$), and -company (BC). For the scope of this work we consider that a business dependency model is created for a single BC. The ordering $BR \prec BF \prec BP \prec BC$ represents the strict topological ordering of graph M. Every edge $E \in \vec{E}$ represents a dependency. Let $V \in \vec{V}$, then let $\vec{E}_V \subseteq \vec{E}$ be the set of edges directed to V, and let $\vec{D}_V$ be the set of vertices from which $\vec{E}_V$ origin, i.e., $\vec{D}_V$ is the set of dependencies of V. For every vertex $V \in \vec{V}$ a conditional probability distribution (CPD) $P(V|\vec{D}_V)$ is given, or, alternatively, a combination function is given for V and edges $E \in \vec{E}_V$ are associated with conditional probability fragments s.t. a $p(+v|d)$ is given for all $d \in \mathrm{dom}(D), \forall D \in \vec{D}_V$.* ▲

Definition 3 solely considers inter-layer dependencies, and excludes intra-layer dependencies, e.g., we exclude dependencies of a business function onto another business function. We argue that such dependencies are resolvable in a lower level and by an adequate specification of associated CPDs. With Definition 3, a mission dependency model represents a probabilistic graphical model, and, in particular, a Bayesian network, as, e.g., defined by Pearl and Russel in [14]. A key feature of Bayesian networks is the ability to locally interpret individual parameters, i.e., to locally interpret individual probabilities of CPDs. This feature allows Bayesian networks to be a direct representation of the world, as stated by [14]. Pearl and Russel call this understandability the "local semantics" of Bayesian networks, which we will preserve for our presented probabilistic mission impact assessment, i.e., to provide a direct and local understandability to all conditional probabilities, e.g., $p(+x|+y) = 67\%$.

The local semantics of mission dependency models, and the direct relationship to commonly known entities in BPMN models, intuitively allows business experts to model (partial) mission dependency models directly. When given information from multiple sources, e.g., multiple BPMN models and information gathered from different experts from different expertises, a common problem is eminent: Experts frequently use different nomenclatures, descriptions, languages and references for common entities, inevitably leading to semantic overlaps between information sources due to semantic and terminology gaps. Therefore, a naive concatenation of extracted and gathered business dependency model inevitably leads to duplicate entities and incorrect impact assessments. To obtain well-defined results, i.e., to obtain a solid and consistent business dependency model from multiple sources and experts, a semantic normalization and merging is required for business dependency model. We deeply discuss and propose a solution to the semantic normalization and merging problem of mission dependency models in [15].

The following example shows the intention of using probabilistic dependency models for mission impact assessments.

Example 1. *Following the mission dependency model depicted in Figure 2 (excluding dashed entities), a Bayesian network is evident representing a joint probability distribution (JPD) over all random variables as*

$$\begin{gathered}P(CM_1, BP_1, BF_1, BF_2, A, B) = \\ P(CM_1|BP_1) \cdot P(BP_1|BF_1, BF_2) \cdot P(BF_1|A) \cdot P(BF_2|A, B) \cdot P(A) \cdot P(B) \ , \end{gathered} \tag{1}$$

i.e., the product of all locally defined CPDs. $P(BP_1|BF_1, BF_2)$ *and* $P(BF_2|A, B)$ *are obtained through the noisy-or assumption from* $p({+bp_1}|{+bf_1})$, $p({+bp_1}|{+bf_2})$ *and* $p({+bf_2}|{+a})$, $p({+bf_2}|{+b})$ *respectively. Due to the absence of global normalization factors in Eq. 1, locally defined CPDs are interpretable for themselves, e.g.,* $P(BF_2|A, B)$ *and respectively* $p({+bf_2}|{+a})$ *are understandable* without *a need to consider the degree of, say,* $p({+bp_1}|{+bf_1})$.

With a sound definition of a JPD, one now, e.g., obtains a probability of impact ${+cm_1}$ *onto the company* CM_1 *from, say, an observed impact on* $A = {+a}$ *and none on* $B = \neg b$ *by probabilistic inference. By marginalization of* ${+cm_1}$ *from the JPD, one obtains the intended result as*

$$P({+cm_1}|{+a}) = \alpha \cdot \sum_{BP_1} \sum_{BF_1} \sum_{BF_2} P({+cm_1}, BP_1, BF_1, BF_2, {+a}, \neg b) \ , \tag{2}$$

with a normalizing factor α, *s.t.* $\sum_{CM_1} P(CM_1|{+a}) = 1$. *Later, we will define exactly this probability of impact onto a company as a well-defined mission impact assessment. The obtained result* $P({+cm_1}|{+a})$, *say, 20% is a plain conditional probability. Moreover, the obtained result* $P({+cm_1}|{+a})$ *is defined to be formally correct, given the defined data (defined dependency models) is validated to be correct. This has the advantage that obtained results can be reported to higher authorities without disclosure or explanation of used algorithms or approaches.* ♦

The example shows how a mission impact can be defined based on a probabilistic inference problem. An obtained result of, say, 20% is understandable without a context, i.e., one does not require indepth knowledge of an originating attack or needs to understand how this assessment is obtained. Further, a probability of 20% is interpretable unbiasedly, i.e., every person should come to a similar conclusion on how likely this assessment is. Depending on a use case, i.e., what is at stake, this likeliness must then immediately trigger a person's situational awareness for the criticality of this situation. Note that, a degree of criticality originating from a probability of impact depends on the given use case environment and is an objective measure. For example, two operators are given assessments of their working environment: a 10% probability of failure of a nuclear power plant and a 10% probability of contamination of a fish farm. Without knowledge of probabilistic inference, involved models, or this article, both experts must come to the same conclusion how likely both impacts are. Moreover, what is at stake in their environment is an objective measure, which to judge lies in their expertise, e.g., a monetary loss. Therefore, both experts are immediately able to interpret obtained impact assessments and their implications unbiasedly without being trained further.

To detail an effect of an impact further, we define a set of *consequences* for every business process to which a possible failure of the business process might lead. Again, a consequence is modeled as an individual conditional probability stating the probability that a consequence happens, given an impact on the business process. Likewise, one can then calculate the probability that a BP's consequence happens ($+con_{BP}$), given all observed local impacts, say $+a$, plainly as $P(+con_{BP}|+a) = P(+con_{BP}|+bp) \cdot P(+bp|+a)$.

Still, only considering a business view does not cover transitively (or passively) involved resources. To cover distant and widespread dependencies, which are not directly obvious, we introduce a network dependency model in the upcoming subsection.

3.2 Resource Dependency Model (Operation View)

As mentioned before, an identified critical resources might be threatened *transitively* by further resources inside an infrastructure, or rather, *depend* on further resources. For example, a business expert might identify a frontend web-server for accessing critical data involved in a business process. This does accurately represent a business perspective, but, from an operational perspective, the web-server is only a small part of a complex process: variously involved databases fed by computational clusters and interfaces are equally, or even more, important. However, deeply understanding all transitively involved resources exceeds the expertise of a business experts and might even be unknown to an IT specialist. Nevertheless, all transitively, indirect and direct resources must be covered in order to provide an accurate mission impact assessment.

Therefore we introduce a resource dependency model covering dependencies between individual resources, which can be, e.g., individual ICT servers, ICS devices, software components or, in other use cases, manufacturing robots, suppliers, soldiers or vehicles. We follow the same probabilistic approach as before, i.e., every dependency between two resources represents a local conditional probability of impact, given the dependence is impacted, as shown in Figure 3. We define a resource dependency model formally as follows.

Definition 4 (Resource Dependency Model)**.** *A resource dependency model R is a directed graph as a pair $\langle \vec{V}, \vec{E} \rangle$ of vertices $\vec{V}$ and edges $\vec{E}$. Every edge $E \in \vec{E}$, from vertex $X \in \vec{V}$ to $Y \in \vec{V}$ represents a dependency, and is associated with a conditional probability fragment $p(+y|+x)$. Vertices $\vec{V}$ are random variables (Def. 1) and represent resources in an infrastructure, where a subset of vertices semantically correspond to vertices of a corresponding mission dependency model M. Let $V \in \vec{V}$, then let $\vec{E}_V \subseteq \vec{E}$ be the set of edges directed to V, and let $\vec{D}_V$ be the set of vertices from which $\vec{E}_V$ origin, i.e., $\vec{D}_V$ is the set of dependencies of V. For every vertex $V \in \vec{V}$ a conditional probability distribution (CPD) $P(V|\vec{D}_V)$ is defined by a non-leaky noisy-or combination of all conditional probability fragments of associated edges in $\vec{E}_V$. V is not contained in $\vec{D}_V$, i.e., a resource V is not dependent on itself.* ▲

The definition of a resource dependency model is similar to the definition of a mission dependency model (Definition 3), and does represent a probabilistic graphical model as well, but does not introduce constraints of acyclicity, i.e., a resource dependency model can contain cyclic dependencies. Therefore, a resource dependency model is not a Bayesian network. We preserve desired local semantics of parameters, i.e., local conditional probabilities, by an exploitation of employed noisy-or combination functions, as described in Section 4.

Using two individual models, one from a business perspective and one from an operational perspective, is highly beneficial, as knowledge from different experts is included directly and is used to accept potential disagreements: If one model had to be agreed on from both perspectives, the identified web-server would be disputed as it is not clear which resources are directly mission critical and to which depth all dependencies have to be covered. In the worst case, too many vaguely relevant resources would be identified, or, too few resources would be identified. By the use of two models, each perspective is correctly represented without a need to make any compromise.

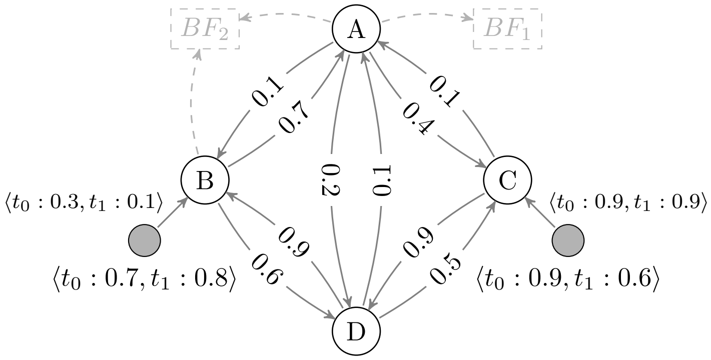
BF_2
A
BF_1
0.1
0.7
0.1
0.4
0.2
0.1
B
C
$\langle t_0 : 0.3, t_1 : 0.1\rangle$
$\langle t_0 : 0.9, t_1 : 0.9\rangle$
0.9
0.6
0.9
0.5
$\langle t_0 : 0.7, t_1 : 0.8\rangle$
$\langle t_0 : 0.9, t_1 : 0.6\rangle$
D

Figure 3: Resource Dependency Model. Dependencies between B, C would also be possible. Conditional probability fragments are marked along the edges. Grey nodes represent external shock events leading to local impacts. The time-varying conditional probability of local impact given an instantiated external shock event is given next to the edge and the time-varying shock event's prior random probability is given below it. Connections to the mission dependency model are sketched in dashed gray.

Often and in contrary to the mission dependency model, assessing resource dependencies is not manageable by hand. Complex operation structures render a manual dependency analysis infeasible and error prone. Further, dynamically adjusting infrastructures (e.g., as found in IT cloud use cases) make it even unknown to an expert to identify exact dependencies. However, we argue that an expert is able to validate a presented resource dependency model for plausibility. Therefore heuristics are employed based on exchanged information amounts, e.g., traffic analyses in an IT use case, to learn possible resource dependencies. As long as a resource only consumes relevant information for its purpose, then every information transfer must motivate some dependency. Moreover, collecting traffic-information is a reasonable and feasible effort. Further, under the assumption of *per node* equally distributed entropy and encoding of consumed information, a dependency, i.e., a conditional probability, must be a function of consumed information bits. We, therefore, reduce an infeasible effort for an expert of identifying all dependencies by hand onto finding a heuristic, or rather, validating of a generated dependency model. While a validation of a resource dependency model is expensive, it is a reasonable effort for highly critical infrastructures or operations. The following example demonstrates an approach for an automatic generation of a resource dependency model.

Example 2. *It is reasonable to assume that it is feasible to acquire information about exchanged information at a logical ICT device level covering virtual machines as individual devices. Often, more granular data, e.g., on software layers, is not acquirable, and exact dependencies between software components are not identifiable. Still, multiple software applications running on one device are very likely dependent on each other, and an impact of one software component will lead to an impact of other software components. We therefore say that dependencies at network device level are coarse enough, and assume that every device drives one purpose that might be fulfilled by multiple software components.*

For example, a workstation X consuming different query results from multiple databases will distribute gained and processed information from such queries to other devices. The percentage of received traffic $T_{Y_i,X}$ from every database Y_i towards the total received traffic gives a good guideline for the conditional dependency between them as $p({+x}|{+y_i}) = \frac{T_{Y_i,X}}{\sum_i T_{Y_i,X}}$. In general, this heuristic should apply to all cases where all dependencies of a resource provide similarly encoded data with similar entropy. However, if the workstation further consumes irrelevant 5TB of cat pictures from a local file server, the heuristic will fail, because the workstation also consumed many irrelevant information. Depending on a network or company characteristics other heuristics might be appropriate, e.g., derivation from a mean received amount of data or a mapping onto a σ distribution. ⧫

In Section 5 we show that the straightforward approach outlined in Example 2 delivers highly satisfying results for two real world use cases. Note that the approach outlined in Example 2 generates one model for the complete period of observed network traffic, i.e., represents a general model of all evolution phases if a network changes over time. By periodically repeating the approach outlined in Example 2, a model incrementally adapts to changing, dynamic environments, and a differential analysis to a fixed time-period reference point (e.g., a monthly generated model) can be used to cope with context drifts. In that sense, the here discussed model can still be used in slightly dynamic environments. Later, in Section 6 we present an approach towards completely dynamic probabilistic mission impact assessments for rapidly changing environments and time-dependent analyses at a, if intended, nearly continuous time granularity.

Mission dependency models and resource dependency models directly represent an infrastructure and already define a probabilistic graphical model. What is yet missing to perform a mission impact assessment using this model is a source of potential impacts addressed in the following subsection.

3.3 Local Impacts (Security View)

A third view involves a security expert able to assess local consequences of events. In the style of reliability analyses using Bayesian approaches we model external shock events inside a network. Informally, an external shock event (SE) represents a source for an impact and is attached to a node in a resource dependency model, i.e., a SE threatens a node to be impacted. By representing SEs as random variables, one gains the ability to include uncertainty about the existence of SEs and uncertainty about whether a present threat leads to an impact on a node. Formally, we define external shock events as follows.

Definition 5 (External Shock Events)**.** *An external shock event SE is a random variable and let $\vec{SE}$ be the set of all known external shock events. An external shock event $SE \in \vec{SE}$ might be present (+se) or not be present ($\neg se$), for which a prior random distribution $P(SE)$ is defined, i.e., SE is a prior random variable. Every vertex V of a resource dependency model R might be affected by one or more external shock events $\vec{SE}_V \subseteq \vec{SE}$. In the case that an external shock event is present ($SE = {+se}, SE \in \vec{SE}_V$), there exists a probability of it affecting node V, expressed as a local conditional probability fragment $p({+v}|{+se})$. If an external shock event exists and it is not inhibited, we speak of a* local impact *on V. In the case that the external shock event is not present, i.e., $\neg se$, it does not affect random variable V and we write $p({+v}|\neg se) = 0$. Every individual conditional probability fragment from an external shock event is treated in the same noisy-or manner as a dependency towards another node, and thus, multiple shock events can affect one node and one shock event can affect multiple nodes.* ▲

According to Definition 5, the presence of an external shock event can be known (observed) or can be unclear and is assessed probabilistically through its prior random distribution $P(SE)$. We denote the set of observed external shock events (known presence) as a set of instantiations $\vec{se}_o$ of observed random variables $\vec{SE}_O \subseteq \vec{SE}$. This is highly beneficial for applications, where the actual presence of impact-sources is uncertain ($P(SE)$), and where evidence of existence and impacts is available, i.e., where SEs are observable (${+se} \in \vec{se}_o$). Classically, a local impact can also be seen as an observation of an impacted node, i.e., $+x$. However, in a Bayesian network approach an observation implies that a reason for the observation is evident from and modeled in the network, i.e., an impact origins from a modeled network and a reason for it lies within an impact of some *other* nodes. By introducing external shock events one gains the ability to model "soft evidence" of local impacts, i.e., one is unsure whether an external shock event exists, and is unsure whether it might actually lead to a local-impact and affect a node's operational capability. Nevertheless, observations provide valuable information about transitive impacts and are further discussed after the following definitions and examples.

To further detail a local impact assessment, we introduce the concept of temporal aspects to represent that impacts do not necessarily remain constant over time.

Definition 6 (Temporal Aspects)**.** *We define a temporal aspect of an external shock event. We employ the idea of abstract timeslices in which the effect of an external shock event changes. Every abstract timeslice then represents a duplicate of the network- and mission dependencies with a different set of local conditional probabilities and prior probabilities of local impacts. We denote time-varying probabilities in a sequence notation as* $\langle t_0 : p_0, \ldots, t_T : p_T \rangle$*, with* $T + 1$ *abstract timeslices. In every abstract timeslice i, varying local impacts take their respective conditional or prior probability* p_i *defined for its timeslice* t_i*.* ▲

With Definition 6 an independent model is created for each timeslice, where impact assessments of time t_i are independent of assessments from time t_{i-1}. In the following, we call this the “independent-timeslice model.” This idea is extended in Section 6 towards a fully dynamic impact assessment, where entities of timeslice t_i *depend* on entities of timeslice t_{i-1}, i.e., a resource dependency model is a time-dependent model evolving over time with time-dependent, “conscious” nodes allowing for retrospect and predictive analysis of potential mission impacts.

Every local impact represents a potential threat and can be, for example, a consequence of a present vulnerability, a countermeasure, a failure or an attack. It lies in the expertise of a security operator to assess a potential *local* impact of those threats. Due to locally viewed CPDs based on combination functions from probability fragments, an expert does neither need to have any expertise in resource dependencies nor an understanding of missions to do so. Further, an assessment of local impact probability can be formally validated through experiments or be grounded on commonsense. The following examples demonstrate how to employ external shock events in an ICT security context and outlines the merits of local assessments.

Example 3 (Response Plan Side Effects)**.** *We employ mission impact assessment to achieve a* qualitative *assessment of potential negative side effects of a proposed response plan to an ongoing or potential attack. We see a response plan as a collection of individual actions affecting a network. E.g., a shutdown of a server might easily reduce the surface of a potential attack. Still, if a critical resource is highly dependent on that server, it might impact a mission even heavier than a potential attack. We consider three mitigation-action types and transform them to external shock events, possibly leading to local impacts.*

The first mitigation action, i.e., an external shock event, is a ***shutdown****. Obviously, if a node is shut down (+se: the external shock event is present) it is easy to see that the probability of local impact, given the shutdown of node* X*, is* 1*, i.e.,* $p(+x|+se) = 1$*.*

Secondly, employing a ***patch*** *on a node* X *might produce collateral damage as well. During installation of the patch, there exists a (low) probability of immediate conflict, e.g., a flat assumption of* 10% *or a measure published by the* *software vendor. In a mean time, a patch might enforce a reboot of a network device. This leads to a temporal shutdown and might lead to hardware failure. Finally, after a successful reboot, a replacement of hardware, and/or a restore of a previous backup, the network device will fully resume its operational capability. Using temporal aspects, one is able to model a patching operation in three abstract timeslices and define the local impact probabilities of this external shock event to be* $p(+x|+se) = \langle t_0 : 0.1, t_1 : 1.0, t_2 : 0.0\rangle$.

Our third considered mitigation action is the restriction of a connection from node X *to node* Y*, i.e., a new* ***firewall*** *rule. From a technical perspective this operation forbids a transfer of data that might have been crucial for the operational capability of a node* Y*. Therefore, a firewall rule leads to an operational impact on* Y*. As a connection between two devices resembles a dependency, one must further actually remove this dependency. Otherwise, one would infer further impacts over a dependency that was prohibited and already assessed locally. To do so, one transforms a prohibited dependency to an observed external shock event* $+se$ *s.t. the local conditional failure probability* $p(+y|+x)$ *becomes a local impact probability* $p(+y|+se)$*. Another approach, decidable by a security operator, would be to accumulate prohibited connections and to add an unified local impact for them.*

As these external shock events are deliberately placed inside our domain, we model their prior probability to exist as a tautology, i.e., $p(+se) = 1$*, and, obviously, fully observe the presence of mitigation actions.* ♦

This examples shows how executed actions are modeled as external shock events for an assessment and is applied in a real world use case outlined in Section 5.2. The assessments of *local* impacts are highly beneficial for the example, as not enough, if even any, data is available that allows for an analysis of potential impacts on a company related to executed individual mitigation actions. Without a context-free and bias-free assessment, one needs to generate data for learning and validation of an algorithm: Every mitigation action, and every mitigation action combination, must be executed on all network resources multiple times and an impact onto a company must be assessed. To obtain statistically sound results, we believe such tests must, at least, be executed several thousand times. Frankly, it is easy to imagine that a company would not exist anymore before experiments have finished. In our approach, only local assessments of mitigation actions are required which are validatable locally by using common sense or by using small local experiments, without a need to validate a global assessment (cf. Section 4). In our approach, if expert validation is not available or seen as not sufficient, for example, the impact probability of a patching operation on a resources can be validated by small local experiments on single, automatically deployed instances of virtual machines, which deploy and execute the patch and evaluate its outcome.

A second use case is motivated from an opposite perspective. While response plans discussed in Example 3 are intentionally executed actions on an environment, i.e., local impacts are triggered internally, the following use case considers impacts triggered by external sources, e.g., an adversary.

Example 4 (Vulnerability Impact Assessments)**.** *In a cyber security context, vulnerability advisories represent notifications of potential flaws in systems or softwares. It is not always known if a vulnerability is actually "exploitable," meaning, if a flaw can actually be exploited to cause harm to a system. Further, the expected amount of potential harm can be difficult to assess and depend on local configurations of components or further environment constraints. Moreover, inferring if a vulnerability is actually present on a local system requires a deep analysis of local software configurations. These facts motivate to model vulnerabilities as (partially observed) external shock events.*

A vulnerability represents one external shock event SE_V*, which affects multiple nodes* $\vec{X}$*. The prior probability distribution of an external shock event* SE_V*, i.e.,* $P(SE_V)$*, then represents (i) a probability of existence* $P(SE_V^e)$*, e.g., it affects all nodes running software* Q*, but it depends on numerous further uncheckable constraints if vulnerability* V *actually affects this configuration and represents (ii) the probability of exploitability* $P(SE_V^x)$ *if a (publicly known) exploit exists for a known vulnerability.* $P(SE_V^x)$ *is very likely to vary over time for which one can employ abstract timeslices. The prior probability distribution* $P(SE_V)$ *is then obtained by* $P(SE_V)_t = P(SE_V^e) \cdot P(SE_V^x)_t$ *and can be extracted from CVSS scores based on access complexity, authentication and access vector attributes, where access complexity is likely to vary, i.e., decrease, over time. Moreover, the prior random distribution* $P(SE_V^e)$ *of each external shock event provides the ability to include uncertainity whether an individual shock event does exist or not.*

Given an exploitable presence of a vulnerability on a node X*, a local impact might be created, i.e.,* $p(+x|+se_v)$*. Likewise, this probability represents the expected harm of a successful exploitation of a vulnerability on a node and can vary for every individual node. A required probability fragment for an* impact $p(+x|+se_v)$ *is obtainable by a noisy-or combination of the CVSS attributes for confidentiality-, integrity- and availability impact.*

All parameters, i.e., $P(SE_V^e)$*,* $P(SE_V^x)$ *and* $p(+x|+se_v)$*, are qualitatively assessable and understandable for an expert, who can be assisted, or even be replaced, by an automatic extraction from public vulnerability advisories. Generally, abstract timeslices could be used to model a vulnerability in different dimensions, e.g., C, I, A. We refrain from this idea and keep the nomenclature of a general* impact *on a node.* ♦

Examples 4 and 3 demonstrate the use of external shock events to include knowledge about a resource's impact from external perspectives. Notwithstanding, similar knowledge is obtained from sensors inside a network, e.g., intrusion detection systems (IDS). Alerts obtained from, e.g., IDS represent internal information and are seen as *observations* of random variables, as discussed above. In order to directly represent this knowledge as external shock events, every raised alert is seen as one external shock event, with a prior random distribution $P(SE_A)$ representing the certainty of the raised alarm and $P(+x|+se_a)$ representing the severity, i.e., most likely 1, of the raised alarm. As discussed above, a raised alert may provide transitive information about other resources as well, and we continue the discussion of observations and their implications in probabilistic mission impact assessments in Section 6.

Modeling vulnerabilities in a probabilistic model is significantly different from existing approaches to include vulnerability advisories to raise situational awareness, as, e.g., generating attack paths (cf. [16, 17]). Attack paths try to address the problem how an attacker might actually compromise the network, i.e., they try to simulate an attacker. In contrary, we intend to raise an amount of situational awareness that provokes a proactive removal of potential impact sources. In fact, examples like StuxNet (see, e.g., [18]) have shown that actual attack paths are only loosely based on an interaction of vulnerabilities, but rather that vulnerabilities represents first stages of attacks. Further, we argue that global effects of local vulnerabilities are not foreseeable by any expert. In our probabilistic model, only local consequences of exploited vulnerabilities must be addressed, and transitive effects are (automatically) assessed due to the resource dependency network. This means, one considers what all could happen *locally* and one does not try to find an actual path of an attack or somehow assess global effects at once. An actual use case demonstration for this example is given in Section 5 and we show that one obtains an assessment that is understandable directly and bias-free, namely "there exists a x% probability of compromise to our company" instead of "there exists an attack path over cve-xy, cve-zt, cve-ix, cve-po," which is only understandable to an IT security expert and whose implications are unclear for any non-security expert with indepth domain knowledge. The mathematical foundations for a context- and bias-free probabilistic mission impact assessment from the defined dependency models are introduced in the following sections.

4 Probabilistic Mission Impact Assessment

One obtains a probabilistic graphical model from a mission dependency model and a resource dependency model. Informally speaking, in the resource dependency model, some nodes are threatened by external shock events, and, as nodes are dependent, a threatened node might again threaten another node, leading to a "spread" of impacts. We say, a node is threatened by an external shock event *transitively.* In the end, there exists a probability that even a business process or the complete modeled company (mission) is threatened transitively by various external shock events. To recall, to be threatened by an external shock event (might) lead to an impact; and it is a well-defined problem of calculating this "might"-probability of being impacted due to an external shock event, which is what we call the mission impact assessment as done in Example 1.

Definition 7 (Mission Impact Assessment, MIA). *Given a mission dependency model M, a resource dependency model R and a set of external shock events $\vec{SE}$, a mission impact assessment of a mission node MN is defined as the conditional probability of a mission node $MN \in M$ being impacted (+mn), given all observed external shock events $\vec{se}_o$, i.e., $P(+mn|\vec{se}_o)$, where the effects of local* *impacts due to all $\vec{SE}$ are mapped globally based on mission-dependency and resource-dependency graphs. Note that $\vec{se}_o$ includes present (+se) and absent (¬se) shock events and that some shock events are unobserved, i.e., are assessed probabilistically through their prior random distribution $P(SE)$. The task of obtaining $P(+mn|\vec{se}_o)$ is defined as the* MIA problem. ▲

Given Definition 7 it is the task of an mission impact assessment to solve the MIA problem, i.e., to obtain the probability $P(+mn|\vec{se}_o)$. From a probabilistic point of view, a sound definition of an overall joint probability distribution as demonstrated in Example 1 is required. As demonstrated, this is given for the mission dependency graph, because it is a directed acyclic graph and represents a Bayesian network. However, in the resource dependency graph an acyclicity constraint cannot be assumed and Bayesian network semantics are not well-defined for cyclic graphs. One could transform a network dependency graph to a Markov random network, which, however, due to a needed global normalization factor, destroys an intended local view, i.e., local semantics, on probabilities. Under an employed noisy-or assumption, one can see the probabilistic model as a probabilistic logic program preserving desired local semantics, where every "path" $w_i^{MN} \in \vec{w}^{MN}$ from an external shock event $SE \in \vec{SE}$ to the mission node MN is a conjunction of Boolean random variables and is a sufficient *proof* for satisfying $\{MN = true\} = +mn$. Under a noisy-or assumptions, $\vec{w}^{MN}$ then represents a disjunction of conjunctions. Every proof w_i^{MN} exists with a probability $P(w_i^{MN})$, where $P(w_i^{MN})$ is the product of all probabilities in this proof. Let $\mathbf{P}(w_i^{MN})$ denote the probability viewed as a set. $P(+mn|\vec{se}_o)$ is then the probability that at least one proof holds, or rather, the probability that the disjunction of conjunctions is satisfied, i.e.,

$$P(+mn|\vec{se}) = \bigcup_i \mathbf{P}(w_i^{MN}) = P(\vec{w}^{MN}) = P(\{\bigvee_i w_i^{MN}\}) \ , \tag{3}$$

where not all $\mathbf{P}(w_i^{MN})$ are disjoint (see Figure 4) and it is worth noting that proofs share common "edges" and end in common shock events. Calculating $\bigcup_i \mathbf{P}(w_i^{MN})$, i.e., an exact solution to the MIA problem, is also known as the probabilistic satisfaction problem and is also used in the Problog reasoning framework [19]. If a mission node MN employs a noisy-and combination function, $\vec{w}^{MN}$ is a conjunction of conjunctions, i.e., a special case of the discussed disjunction of conjunction case with only one disjunction, which is straightforwardly to implement, as described in the following section. Note that, as every edge represents a conditional probability and a shock event is a prior random variable, plain summation would double count these probabilities and lead to spurious results. This is exactly the issue from which many fudge-factor based "propagation" algorithms in ad-hoc solutions suffer: If probabilities, i.e., parameters, are counted multiple times, they are not interpretable locally anymore, as it is unknown in advance how and how often parameters will be used by a propagation algorithm, which, moreover, is a specific property of each novelly created propagation algorithm.

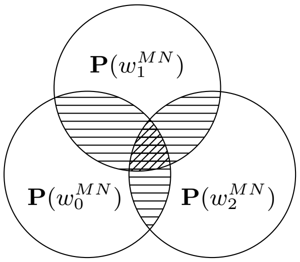

Figure 4: Illustration of $\mathbf{P}(w_i^{MN})$ viewed as sets. Overlapping parts (filled with patterns) are commonly shared probabilities between proofs and are not allowed to be counted twice (or even multiple times) when calculating $\bigcup_i \mathbf{P}(w_i^{MN})$.

The defined probabilistic mission impact assessment $P(+mn|\vec{se}_o)$ directly originates from the definition of all defined dependency-models and represents an inference problem in a probabilistic graphical model. Therefore, all parameters, i.e., probabilities, of defined dependency models remain interpretable locally. Moreover, if locally defined dependency-models are validated to be correct, an obtained impact assessment $P(+mn|\vec{se}_o)$ is *validated, too.* This means that only locally defined parameters and *not* a complete mission impact assessment must be validated by an expert or against ground truth. Moreover, as $P(+mn|\vec{se}_o)$ is a conditional probability, it is bias- and context- free understandable and is therefore more suitable for reporting along a chain of command.

An exact solution to the MIA problem by a calculation of $\bigcup_i \mathbf{P}(w_i^{MN})$ is possible by the inclusion and exclusion principle and the Sylvester-Poincaré equality. Still, calculation is exponential in the number of proofs due to the subtraction of all overlapping sets and is therefore not practical. We therefore approximate a solution by the use of a Monte-Carlo simulation.

4.1 Monte-Carlo Approximation

To find an approximate solution to the MIA problem, we use a two step approximation procedure.

For every mission node MN in a mission dependency model M, there exists a Boolean formula $\vec{w}^{MN}$ as a disjunction of conjunction over Boolean random variables $\vec{B}$. However, Boolean random variables in $\vec{B}$ take their respective truth value according to a probability distribution. As every conjunction in a disjunction is sufficient to proof $+mn$, we speak of them as probabilistic proofs.

Definition 8 (Probabilistic Proofs)**.** *For every business resources $BR_i \in \vec{BR}$ let $\vec{w}^{BR_i}$ denote the set of all proofs and let $w_j^{BR_i}$ denote the j^{th} proof. Let $\vec{w}$ denote the super-set of all found proofs. Every proof $w_j^{BR_i}$ is a set of individual conditional probability fragments $p(+x|+y)$, representing an edge, i.e., a dependency of X on Y. The product of all probability fragments $p(+x|+y) \in w_j^{BR_i}$ is the satisfaction-probability of a proof $P(w_j^{BR_i})$. Every proof $w_k^{BR_i}$ for which holds*

$\exists j : w_j^{BR_i} \subseteq w_k^{BR_i}$ is irrelevant for calculation and $\vec{w}$ is a finite set. Informally this means, during proof search along one "path" an already visited node must not be visited again and we cannot get stuck in infinite loops. ▲

Based on a set of probabilistic proofs, one obtains an algorithm to find an approximate solution to the MIA problem for all nodes of a mission dependency model.

Definition 9 (Algorithm to the MIA Problem)**.** *Let M be a mission dependency model (Def. 3) consisting of business resources $\vec{BR}$, business functions $\vec{BF}$, business processes $\vec{BP}$ and a business company BC. Let R be a resource dependency model (Def. 4). Let $\vec{SE}$ be a set of external shock events (Def. 5) affecting resources in R and let $\vec{SE}_O = \vec{se}_o$ be a set of observed shock events.*

Then, for every business resource $BR_i \in \vec{BR}$, $\vec{w}^{BR_i}$ is obtained by a depth-limited search from BR_i through R to any $SE \in \vec{SE}$. A complete sample $\vec{s}$ of all random variables in $\vec{w}$, M and $\vec{SE} \notin \vec{SE}_O$ is drawn according to each respective random distribution, and observed random variables $\vec{SE}_O$ are instantiated to their observed value in $\vec{se}_o$. Consecutively, for every BR_i the satisfaction of $\bigvee \vec{w}^{BR_i}$ by $\vec{s}$ is checked and marked on BR_i by a respective $+br_i$ or $\neg br_i$. Subsequently, for every noisy-or $BF_i \in \vec{BF}$ the satisfaction of all $\bigvee_{BR_i} p(+bf_i|+br_i) \wedge BR_i$ by $\vec{s}$ is checked and marked accordingly. Respectively, for every noisy-and $BF_i \in \vec{BF}$ the satisfaction of all $\bigwedge_{BR_i} p(+bf_i|+br_i) \wedge BR_i$ by $\vec{s}$ is checked and marked accordingly. Respectively for every $\vec{BP}$ on each $\vec{BF}$ and BC on every $\vec{BP}$. Every satisfaction of a mission node $MN \in M$ is counted by hit_{MN}. Sampling and checking is iterated n_S times. ▲

Theorem 1 (Solution to the MIA Problem by Algorithm of Definition 9)**.** The algorithm described in Definition 9 approximates a solution to the MIA problem on every mission node $MN \in M$ using n_S iterations by $P(+mn|\vec{se}_o) \approx \frac{hit_{MN}}{n_S}$. For a depth-limited search, the algorithm scales linear with the number of edges in R, $|\vec{SE}|$, $|\vec{w}|$ and n_S. For an infinite number of samples n_S and an unlimited depth-search, the algorithm generates an exact solution to the MIA problem. ▲

A proof is given and empirically evaluated in the following section, after a short demonstration of the approximation procedure.

Example 5. *Consider Figure 3, where an identified mission critical resource A (compare Figure 2) is threatened (transitively) by local impacts on nodes B and C. Say, both external shock events are observed to be present, i.e., $\vec{se}_o = \langle +se_B, +se_C \rangle$. We exclude the dependency of BF_2 on B and temporal aspects for brevity. Through depth-first search one finds proofs $\vec{w}^A$ as*

$$\begin{aligned} w_0^A &= \{p(+a|+b), p(+b|+se_B)\} \\ w_1^A &= \{p(+a|+b), p(+b|+d), p(+d|+c)p(+c|+se_C)\} \\ w_2^A &= \{p(+a|+c), p(+c|+se_C)\} \\ w_3^A &= \{p(+a|+c), p(+c|+d), p(+d|+b), p(+b|+se_B)\} \end{aligned} \tag{4}$$

Additional proofs, e.g., $w_{\emptyset}^{A} = \{p(+a|+b), p(+b|+c), p(+c|+b), p(+b|+se_B)\}$, are redundant, as, here, w_0^A is always (already) satisfied, if $w_{\emptyset}^{A}$ is satisfied. After finding these proofs, finding proofs to higher nodes in a mission dependency model, say, to BF_1, is trivial, by simply appending $p(+bf_1|+a)$ to every proof of A. Subsequently, the same holds for BP_1 and CM_1.

For simulation, at first every used random variable is sampled. Let $\vec{RV}$ be the vector of all random variables included in all proofs, i.e., $\vec{RV} = \langle p(+a|+b), p(+b|+se_B), p(+b|+d), p(+d|+c), p(+c|+se_C), p(+a|+c), p(+c|+d), p(+d|+b), p(+bf_1|+a), p(+bf_2|+a), p(+bp_1|+bf_1), p(+bp_2|+bf_2), p(+cm_1|+bp_1)\rangle$. Let $\vec{s}$ denotes a sample of $\vec{RV}$, e.g., $\vec{rv} = \langle +,+,+,+,+,\neg,\neg,\neg,+,+,+,+,+\rangle$, where + represents a true sample, and ¬ a false sample.

Subsequently, for every identified critical resource, i.e., A, one checks if at least one of its proof is satisfied, i.e., if $\bigvee \vec{w}^A$ is satisfied. One obtains that w_0^A is satisfied, which satisfies A. The circumstance that w_1^A is also satisfied, but w_2^A and w_3^A are not satisfied is irrelevant (noisy-or assumption) and further checks can be skipped. Subsequently, one checks the remaining mission dependency graph for further satisfactions in this sampling round. As A and $p(+bf_1|+a)$ are satisfied, BF_1 is satisfied (marked by $+bf_1$) as well. The same holds for BF_2. Likewise, BP_1 is satisfied as well as CM_1. Every satisfaction is marked as a successful Monte-Carlo round and increments a mission node's MN hit counter hit_{MN}.

This procedure is repeated n_S times, i.e., $\vec{rv}$ is sampled and $\vec{w}$ is checked. Finally, every impact assessment of a mission node MN, i.e., $P(+mn|+se_C, +se_B)$, is approximated by $P(+mn|+se_C, +se_B) \approx \frac{hit_{MN}}{n_S}$. ♦

This example demonstrates the approximation procedure on the running example shown in Figure 2 and 3 for two observed SEs. If the SEs are unobserved, their existence must be assessed probabilistically by appending their prior random distribution to every proof in which they are included, e.g., $w_0^A = \{p(+a|+b), p(+b|+se_B), p(+se_B)\}$.

For implementation of the complete procedure, some remarks are made on optimizing the procedure:

Remark 2 (Proof Check)**.** *Checking all proofs during one Monte-Carlo round is highly optimizable. $\vec{w}^{BR_i}$ can be sorted descending by $P(w_j^{BR_i})$ s.t. most likely holding proofs are checked first and subsequent checks can be skipped once a satisfied proof is found. Further, a proof $w_j^{BR_i}$ can be sorted ascending by its individual local conditional probability fragments s.t. most unlikely random variables are checked first and further checks inside one proof can be skipped. Further, a proof w with $P(w) < \frac{1}{n_S}$ will statistically never be drawn, i.e., it can be ignored during simulation and check.* ▲

Note that if large quantities of low-probability proofs are ignored, an error might accumulate leading to slightly less accurate results compared to the evaluated results in the following section.

Remark 3 (Temporal Aspects Implementation)**.** *We introduced that an external shock event, can have different conditional local probabilities depending on an* *abstract timeslice. This means a proof $w_j^{BR_i}$ contains varying probabilities. Naively, one could perform a Monte-Carlo simulation for every abstract timeslice through complete duplication. However, this would redundantly simulate all non-varying probabilities. We therefore partition $w_j^{BR_i}$ in a non-varying set of conditional probabilities, i.e., a "path" to an impacted node, and a set of varying conditional probabilities, i.e., a set of local impacts.* ▲

4.2 Complexity Analysis and Experimental Evaluation

As a central theme, we focus on actual feasibility of our proposal and we demonstrate that our approach scales well, e.g., linearly, with a graph's complexity. In the following, we give a short expected summary of the complexity of our approach and evaluate it experimentally. We provide an experimental validation of expected time-complexities, as well as a verification of the introduced approximation algorithm and empirically prove Theorem 1.

Evaluation and demonstration of the computational complexity of our presented approach is challenging, as it depends on the graph structure of the network and the processed response plan. We therefore use random graphs containing n_N nodes and $n_E = n_N^2 \cdot 0.1$ edges while assuring that every node is at least bidirected. By doing so, one obtains a fully connected graph with, approximately, a 10% chance of two nodes being directly connected. In every experiment, we process $n_{SE} = n_N \cdot 0.1$ randomly placed external shock events, i.e., 10% of all nodes are possibly impacted. We measure the time t_{ps} required for finding all n_P proofs up to depth d_{max}, and t_{sim} required for simulating all found proofs n_S times. Every experiment is repeated in 50 random graphs.

Complexity is differentiated between both steps of the approximation procedure. Given a constant maximum search depth d_{max}, depth-limited search (DLS) scales linearly with the number of edges n_E, as every edge is solely visited once and experimentally evaluated in Figure 9. Further, DLS scales slightly with the number of placed local impacts n_{SE} (compare Figure 13), as a pre-computation of shortest distances to local impacts per node can eliminate dead-ends early. We write, the time required for a proof search is a function proportional to $t_{ps} = f(n_E, d_{max}, n_{SE})$.

Remark 4 (Maximum search depth)**.** *DLS scales exponentially with a specified maximum depth d_{max}. In general and for our example, the maximum depth should be chosen in the range of the average path length inside a given graph s.t. almost every node is considered at least once. In order to better scale with maximum depth it is reasonable to allow a rational d_{max}, where a depth $0 < d_{dec} < 1$ resorts to the best d_{dec}, i.e. most dependent, children.* ▲

Monte-Carlo simulation, i.e., sampling and checking of proofs, scales linearly with the number of found proofs n_P (compare Figure 10) and the number of Monte-Carlo samples n_S per business resource (compare Figure 11), i.e., $t_{sim} = f(n_P, n_S)$. Naturally, the number of proofs n_P scales with the number of local impacts n_{MA}, of edges n_E and the maximum proof length d_{max} (compare Figure 12), i.e., $n_P = f(n_{SE}, n_E, d_{max})$.

In order to verify the correctness and convergence of the proposed Monte Carlo approximation procedure (see Section 4.1), Figure 15 and 14 show the absolute error of the proposed Monte Carlo approximation towards an exact solution based on the inclusion- and exclusion principle. Experiments were evaluated on 100 different networks and convergence results where considered on up to $n = 5\,000\,000$ samples. A mean of an absolute error $\bar{E}$ of the proposed approximation follows $\bar{E}(n_S) = 0.2\sqrt{n_S}^{-1}$ for n_S samples and corresponds to an analytically derived error behavior (cf. [20, Sec. 2.2]), which is expected to be proportional to $\sqrt{n_S}^{-1}$. The variance of the absolute error σ_E for n_S samples follows $\sigma_E(n_S) = 0.17\sqrt{n_S}^{-1}$.

These evaluations greatly prove the characteristics of the approximation algorithm, given in Definition 9, as stated by Theorem 1.

Proof of Theorem 1. All evaluations on scalability, i.e., evaluations in a total of 400 random graphs and a generation of 3600 data points, show an expected linear scalability of the independent-timeslice mission impact assessment with each identified parameter in Theorem 1. All evaluations on the inference accuracy of the approximation procedure, i.e., evaluations in a total of 200 random graphs and a generation of 35 000 data points, show an expected behavior of convergence towards an exact solution to the MIA problem. □

In summary, these experiments show that the discussed approach for an analysis of mission impact in fixed, independent timeslices scales linearly with all input parameters, and solving the MIA problem remains tractable even in largely scaled domains. In the following section, we discuss and highlight the benefits and applicability of probabilistic MIA in two real world use cases involving multiple experts from different domains.

5 Use Case Experiments

In Examples 3 and 4 we discuss two application fields for the introduced probabilistic mission impact assessment. In this section we apply both to two real world use cases involving business-, IT-, and security experts, and discuss and present results for obtained mission dependency models, automatically learned resource dependency models and evaluated mission impact assessments. In summary, the approach shows to be directly applicable, delivers satisfying results and is greatly accepted by experts.

5.1 SMIA Challenge

In 2011 Teodor Sommestad et al. [21] conducted an experiment at the information warfare lab of the Swedish defense research agency, which gives us the opportunity to demonstrate Example 4 for vulnerability impact assessment. In the 2011 experiment, codenamed SMIA2011, a complete network consisting of multiple domains was set up containing multiple ICT servers, clients, mailservers, firewalls, ftps, webservers, even SCADA server, etc. User behavior was simulated inside each domain by action scripts, e.g., checking webservices, emails and downloading files. Intrusion detection systems provided information to one team in charge of monitoring the complete network and noting suspicious behavior. Another team was in charge of infiltrating the network. Both teams carefully documented their approaches and all network traffic was recorded over six days.

In summary and most noteworthy, the attacking team was able to ex-filtrate all mail messages and change parameters of a SCADA server. For our analysis we consider these servers as mission critical resources. While some attacks built on each other, the most impacting attacks were almost uncorrelated. Further, vulnerabilities actually played an insignificant role during attacks. As our approach does not build up on actual attack sequences, but rather considers vulnerabilities as points of interest, we evaluate if our approach is able to raise a significantly high enough situational awareness to be concerned about a compromise, i.e., impact, on the identified mission critical systems.

In more detail, the attacking team firstly discovered a misconfigured service running on one mailserver m_1, which allowed the attackers to extract a (encrypted and later decrypted) password file. By decrypting the password file from m_1 the attackers further gained a privileged ssh connection on m_1, which was exploited for tunneled attacks to two hosts f_6 and o_6 that were not reachable previously. Exploitation of a same known vulnerability on f_6 and o_6 gave a remote shell that revealed further user-passwords. The extracted passwords allowed downloading all mailboxes from all domains, most likely due to reused passwords in multiple domains. Another vulnerability was exploited directly on another host h_a. While revealing new passwords, no further attacks built up on it. Nevertheless, it could have been an excellent starting point for the following and most interestingly attack: The attacking team was able to completely manipulate all employed firewalls, providing complete access to any node on any domain. Firewalls were only secured by a simple password (“password”), but it could have been extracted in one of the previous attacks. Due to the broadened reachability, the attackers had free access to a (otherwise completely unsecured) SCADA server, on which they successfully changed various parameters. Note that most attacks were not executed through any publicly-known software-vulnerability and, therefore, would not have been detected by any analysis of such vulnerabilities alone. In fact, no publicly-known software-vulnerabilities were found on any mission critical resource, and no exploits were launched against them. Achilles' heel lied in a set of badly configured resources and a false sense of security in a presumably demilitarized zone. We believe that it is impossible to accurately identify and devise these chain of events beforehand and to detect these configuration flaws in any automatic or manual expert-driven approach.

To obtain a resource dependency model, we follow Example 2, but only use recorded traffic of the first day prior to any severe actions carried out by the attacking team (such as flooding the IDS and changing firewall configurations). An obtained dependency model as shown in Figure 5 seemed plausible, but dependency degrees showed up to be imbalanced: The amount of analyzed traffic of the first day was short and user scripts did not generate a realistic amount of traffic. Further, no operator was simulated to control or monitor the SCADA server, which is why it did not appear in the dependency model analysis. To overcome these circumstances, a minimal dependency probability of 5% is assumed and the SCADA server is manually modeled to be dependent on an operator from the same domain and vice versa. Note that these manual corrections are exactly foreseen in our dependency model. A domain expert is assisted by a heuristic delivering a locally interpretable model, which is, if needed, corrected and consecutively validated in his expertise.

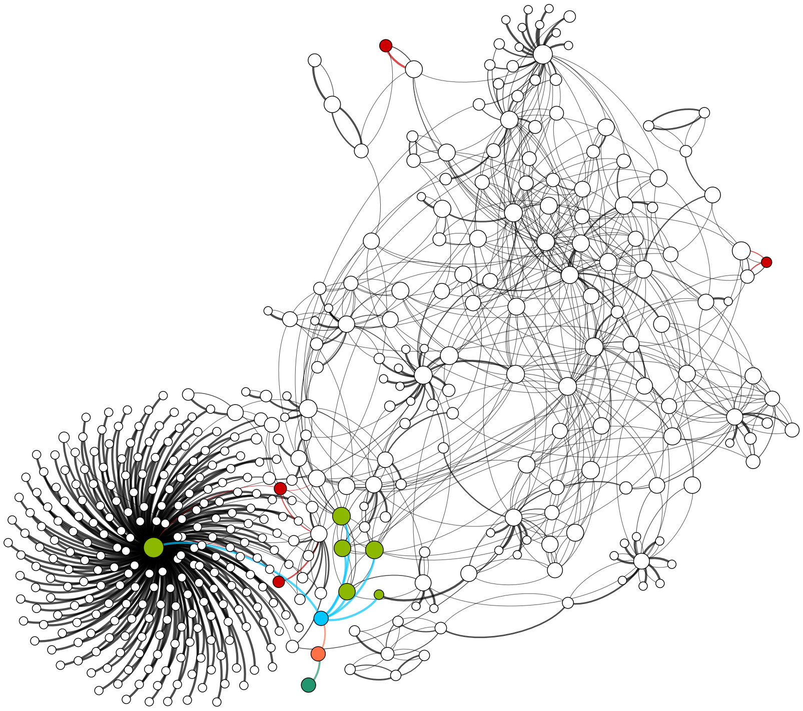

Figure 5: Resource dependency model extracted from one day traffic captures for the SMIA2011 challenge use case. Red nodes are directly impacted by vulnerabilities and affect other nodes transitively (first-step edges highlighted in red) and are remotely placed from mission critical devices (green). Thicker and darker edges represent higher dependency degrees. For visualization some insignificant dependencies are not displayed. In fact, the generated network is far from a fully meshed network (only 1% of all possible edges are extracted.) Number of nodes $n_N = 475$, number of edges $n_E = 2431$. Visualized using Gephi [22].

External shock events are modeled as described in Example 4. Attackers exploited three different vulnerabilities on four hosts (shown in red in Figure 5). Three of these compromised hosts are part of the domain containing m_1 and one is part of the domain containing scada server f_6. For all vulnerabilities, exploits are publicly known and integrated into various frameworks, e.g., metasploit, which is why we assume $P(+se_V) = 1$ for all of them. Local impact probabilities are adapted frankly according to their respective CVSS score divided by 10.

Based on the defined local impacts, the resource dependency model and the mission dependency model, we obtain that there exists a probability of 23.4% of impact, i.e., compromise, of the firstly attacked mail server m_1 and that there exists a 7.8% probability of compromise of the discussed SCADA server f_6. Both servers were in fact compromised, but through ways unforeseeable by any software-vulnerability focused analysis. We argue that a probability in these ranges cannot be ignorable by any person confronted with these results. Even the other compromised mail servers, part of domains without present shock events, are assessed to be impacted with probabilities of 8.1%, 8.1%, 10% and 7.6%. This shows that our MIA delivers reasonable and accurate results, as one obtains nonnegligible impact probabilities raising one's situational awareness for all six servers that were compromised.

Considering all critical resources to be equally part of one business process of a, say, cloud company, an impact probability of 38.3% on the company's mission is assessed. A probability in this range is not dismissable in any case and does not require reference results of previous impact assessments, e.g., it can be directly compared to tossing a 1 or 2 on a six-sided dice. Additionally, this mathematically grounded probability directly measures the criticality of this situation: with an objective measure of this cloud company, e.g., a monetary value expressing the value of the cloud company, one obtains an expected (monetary) loss for this situation originating from this probability of impact.

Moreover, this approach is completely transparent and understandable throughout. Every defined parameter can be grounded on expert assessments or historical evaluations and, finally, a produced assessment of "Due to a set of local, widespread impacts, there exists a probability of 38.3% that our mission will be compromised" is understandable and not dismissible. We argue that this assessment is eligible for a reporting throughout a command-chain and understandable by every instance. In contrary, a non-bias- and non-context-free assessment of, exaggeratedly said, "vulnerabilities lead to a mission impact of 25764.324" does only make sense for a deeply trained expert or given reference results and is not suitable for any reports.

5.2 Acea ARETi Use Case

The Panoptesec integrated research project aims to "deliver a beyond-state-of-the-art prototype of a cyber defence decision support system" [23]. As part of this project, we employ the derived probabilistic mission impact assessment for both use cases as outlined in Example 3 on response plan assessments and Example 4 for a vulnerability impact assessment. We are able to apply and test the complete approach in the Panoptesec's use case partner, Acea SpA, Italy's largest water services operator and one of the largest energy distribution companies in Italy [24]. To deploy a probabilistic mission impact assessment, all three models were created in cooperation with experts, as presented in the following.

In a team session of two business experts from the company and one IT specialist, a complete mission dependency model was identified for Acea ARETi in less than three hours. ARETi is a division of Acea SpA in charge of distributing and controlling energy to the city and vicinity of Rome. Admittedly, business experts showed to be reluctant to give assessments of CPDs, but intrinsically were able to understand all parameters. Given a set of choices, experts quickly agreed on an assessment and validated a complete mission dependency model, which is displayed in Figure 7 in its general structure in combination with the resource dependency model.

The most important objectives, i.e., business processes, of ARETi are to distribute and control high level voltage, whose dependency decompositions are shown in Figure 6. These business processes require four business functions provided by 16 mission critical resources. For distribution of high level voltage (business process BP_D), remote terminal units (RTUs) are required, which are remotely placed actors for switching power. Around 50 RTUs are geographically scattered in the vicinity of Rome and accessed via various communication links, e.g., by GSM. Therefore, these communication establishing devices (further called proxies) as well as the individual RTUs provide business function BF_R. Naturally, a high dependence between RTUs and their proxies exists, which is automatically learned and assessed in the resource dependency model (compare the two "clouds" of devices in the lower right of Figure 7). Therefore, it is sufficient to solely identify involved proxies (i.e., central nodes of the clouds in Figure 7) in a mission dependency model, and not all individual RTUs need be identified. This is highly beneficial for this application, as individual RTUs are frequently replaced, which is automatically captured by an incremental re-learning of the resource dependency model and, therefore, must not be made explicit in the mission dependency model. Two of these proxies are optional and were assessed with a probability fragment of 0.8 each. Remaining proxies are required by business function BF_R with a probability fragment of 0.9 each.

The central intelligence between controlling (business process BP_C) and distributing power (BP_D) is provided by multiple SCADA servers (business function BF_S), which control and manage the individual RTUs, which are monitored by human machine interfaces (business function BF_H). Note that it is sufficient to solely identify the HMI clients (two in this case) in the HMI business function, as it is the directly critical device for the purpose of HMI; transitive dependencies of the HMI clients on SCADA servers, RTUs and proxies are automatically covered through the resource dependency model and probabilistic inference. As both HMI clients allow almost full control over the complete process, both were assessed with a conditional probability fragment of 0.9. All SCADA servers of business function BF_S are laid out redundantly, which is replicated in the mission dependency model by using a noisy-and combination function. As controlling requires a working ICT environment, business experts identified an "ICT" business function BF_I consisting of one NTP-, three archival FTP-, and one web-server. As these devices are not dramatically critical to provide this business function, their probability fragments were assessed to 0.4. BF_I is only marginally required for controlling BP_C and was therefore assessed with a probability fragment of 5%. Dependency degrees for remaining business processes and functions, i.e., conditional probability fragments, were assessed as shown in Figure 6. Further details on these business processes, such as further redundancies and objectives of individual resources are omitted here for confidentially, and only shown in their general structure in Figure 7. In Figure 7 two other business processes are indicated with additional business functions, which are not discussed here for the same reason.

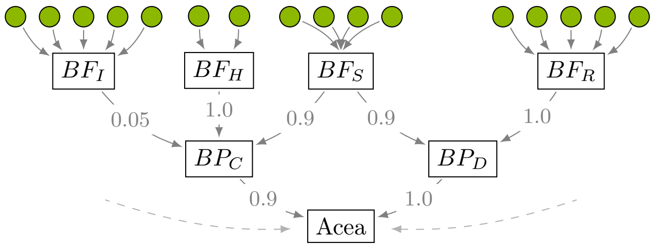

Figure 6: Small exert of two business process involved in distributing (BP_D) and controlling (BP_C) high level voltage in the vicinity of Rome by ARETi and their four required business functions (BF_H, BF_S, BF_R, BF_I) provided by 16 mission critical devices (green). Business resources providing BF_S are laid out redundantly. Compare with the corresponding resource dependency model and full mission dependency model shown in Figure 7.

Further, in the team session, problems outlined in Section 3.1 were evident, where different experts, in fact, used different nomenclatures and languages to refer to the same entities. For example, some experts referenced resources by hostnames, whereas others used IP addresses or some referenced mission nodes using abbreviations and others used literary descriptions. We deeply discuss these challenges of an eminent semantic normalization and merging problem of mission dependency models in [15].

A resource dependency model, depicted in Figure 7 was automatically extracted from recorded traffic in a redundant, backup environment with emulated behavior of SCADA devices as outlined in the Example 2. The extracted model was validated by an external IT specialist consultant to the company to be reasonable. However, we admit that not all individual probability assessments were validated, but critical dependencies were validated to be included and to bear a reasonable dependency degree. As discussed above, large amounts of RTUs are present requiring a remote connection. Therefore, dependencies on routing equipments are critical in this domain as well. In order to cover these routing resources, traffic is not solely analyzed on a logical level, but also on a physical level. To be precise, every connection is established between two logical devices, e.g., identified by two IPs, through two physical devices, e.g., identified by two MAC addresses. Through the use of a global inventory of all

IT related resources, IPs and MACs are mapped towards unique identifiers and a dependence of each resource on its communication-establishing device is added. For example, say, a traffic recording includes a connection from MAC_1, IP_1 to MAC_2, IP_2. Say IP_1 maps to ID_1, IP_2 to ID_2 and MAC_1 to ID_3, MAC_2 to ID_4; then a dependency of ID_2 on ID_1 is added, as well as a dependence of ID_1 on ID_3 and of ID_2 on ID_4. By doing so, one considers that a potential impact on a router, directly affects all resources communicating over said router.

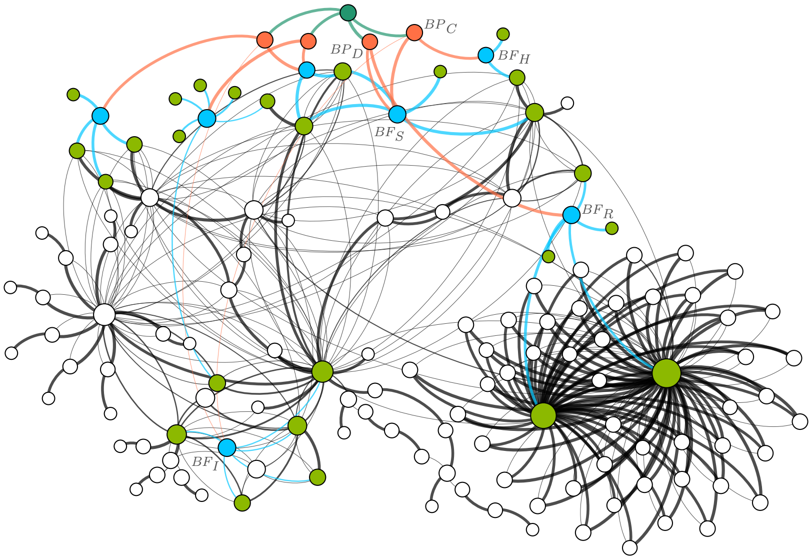

Figure 7: Resource dependency model extracted from roughly one month of traffic captures in ARETi (represented in dark green), where related critical devices are highlighted in green, business functions in blue, and business processes in orange. This model was validated and verified to be reasonable by the company's IT experts. $n_N = 344$, $n_E = 754$. Visualized using Gephi [22]. Corresponding entities to Figure 6 are marked by the required business functions.

In addition to a vulnerability assessment, response plan assessments are highly important for Acea and the Panoptesec project. Assessing how responses to cyber attacks affect an environment is a completely novel problem and no large datasets are available and one must rely on a validation by an expert. In fact, the security specialist and product owner of the Panoptesec system validated the impact assessments outlined in Example 3. As part of the Panoptesec system, response plans are proposed automatically by different components in different situations. Response plans are proposed based on their effectiveness against attacks and financial benefits, which are both not necessarily in line with an assessment of operational impact. For example, a shutdown of highly critical node will certainly eliminate all attacks targeted towards that node and is financially highly attractive, as this response plan does not involve almost any cost. However, this response is catastrophic when considering implications on the mission, i.e., company, as performed by Example 3. In our work by [25] we demonstrate an approach to unify a multi dimensional assessments of response plans and propose a selection of optimal response plans based on an unweighted best compromise in all dimensions. In [25] we evaluate the suitability of the presented mission impact assessment for operational impact assessment to obtain adequate responses to cyber attacks in the here-discussed ARETi environment. In our expertise, we obtained that operational impact assessments of individual response plans bear reasonable assumptions, i.e., venturesome response plans were assessed to bear a high operational impact, and that a combination of impact assessment from financial and operational impact aspects delivers appropriate and non-trivial response plans.

5.3 Intermediate Conclusion

In summary, the presented independent-timeslice probabilistic mission impact assessment allows for causal, context-free, and validatable assessments of parameters and delivers context- and bias-free understandable assessments. Experiments on real data in two real world use cases show that the approach is well accepted, delivers satisfying results, and that all required models are directly obtainable. Moreover, we experimentally and theoretically evaluate that the approach scales linearly even in large domains by the use of a verified approximation procedure.

Still, temporal aspects motivate that assessments require a consideration of *time*. To a limited degree, temporal aspects implement a consideration of time in the presented mission impact model, and, by an incremental relearning of a resource dependency model, dynamic environments are covered as well. However, we intend to consider the following situation as well: A node is threatened due to a set of vulnerabilities, is potentially impacted through them, and dependent nodes become impacted transitively as well. For example, a hypothetical attacker accessing a compromised resource is probably able to extract valuable information passively, able to manipulate processed data, or is able to gain further access through unforeseen events, to and from any dependent resource, i.e., an impact "spreads." If one now assures that the initial resource is not impacted, e.g., by reinstalling the directly affected system, spread *transitive impacts persist and do not vanish* immediately. If an initial impact-source is eliminated, previously caused harm by a hypothetical attacker does not vanish magically, but does persist, i.e., any probability of inferred potential impact must persist to a certain degree over *time*. In the same sense as persistence is "added" to Bayesian networks by dynamic Bayesian networks, we extend probabilistic mission impact assessment to dynamic mission impact assessment in the following section.

6 Extensions to Dynamic Mission Impact Assessment

Temporal aspects introduced in Definition 6 introduce a need for mission impact assessments over *time*. Further, in rapidly changing environments, i.e., dependencies of resources change rapidly over time, require that a mission model is able to be evaluated over time. An extension of Bayesian networks (compare Section 3.1, a mission dependency model is a Bayesian network) towards dynamic domains considering evolutions of states over time is commonly known as a dynamic Bayesian network (DBN). In DBNs values of random variables depend not only on current influences, but also on their respective history. An extension towards a dynamic probabilistic graphical model for dynamic mission impact assessment (DMIA) proposed in this section allows for time-dependent impacts, e.g., decaying impacts, evolving mission impact analyses, and retrospective considerations of potential sources of impacts *inside* a network allowing for forensic and predictive analyses.

So far in this article, we presented how mission dependency models, resource dependency models and impact models are obtainable, validatable and combinable towards one probabilistic graphical model, where individual timeslices are independent. Let R be a resource dependency model, then it is a straight forward extension to introduce a dimension of time t into a time-*dependent* model representing a resource dependency model R^t for each timeslice t. In every R^t, each resource node $RN_i^t \in R^t$ is dependent on its predecessor $RN_i^{t-1} \in R^{t-1}$. With such a transition model $P(R^t|R^{t-1})$ of timeslices, i.e., a model for an evolution of R^{t-1} to R^t, a dynamic probabilistic graphical model (DPGM) as shown in Figure 8 is formed. In a DPGM, nodes, i.e., random variables, are conscious and "do not forget" impacts of preceding timeslices. Respectively, at every timestep t a mission dependency model M^t is formed, whose business critical functions are dependent on some business critical resources in R^t. Note that a mission dependency model M^t is directly equivalent to a previously discussed and derived mission dependency model. Likewise, at every timeslice some resources in R^t are threatened directly by some observed or unobserved shock events in SE^t. However, as described in Section 4 a well-defined semantics is required for such a DPGM, which, due to the (potentially) cyclic nature of a resource dependency model, are not immediately given by classic DBN semantics. Given well-defined semantics, impact assessments at every timeslice are directly obtainable through probabilistic inference, which bears significant advantages, as discussed shortly.

A resource dependency model is derived from (automatic) analyses of communications between resource nodes. Considering such a dependency model over time allows for directly modeling each communication at time t as a dependency, i.e., an influence at time t. This means, with every communication there exists a probability of impact, and a model adapts "live" to ongoing communications. For example, every time a significant amount of data is transferred from a, say, integrity impacted node to another, there exists a probability that the other node becomes impacted as well. We believe that the latter probability

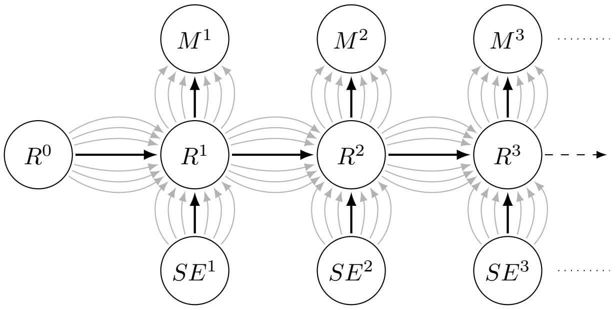

Figure 8: An extension of the presented probabilistic graphical model for mission impact assessment towards a dynamic assessment, considering history states of (potentially impacted) nodes. This is beneficial for applications, where interactions of nodes are rapidly changing over *time* and a live tracking of impacts is required. Resource nodes RN_i^t of a resource dependency model R^t are dependent on their predecessor $RN_i^{t-1} \in R^t$. R^0 represents an initial assumption about the potential impact state of nodes.

is directly equivalent to the derived probability (fragments) as given by the time-independent resource dependency model R, i.e., R is directly reusable for $R^t, t > 0$. R^0 is used to model initial assumptions about impacts present on all resource nodes at time 0, i.e., consists solely of prior random variables.

Still, at a high rate of communications between of nodes as, e.g., present in IT infrastructures, probabilistic inference in DPGMs becomes computationally expensive. Considering an IT infrastructure as an information processing chain, it is reasonable to aggregate communications over a reasonable timeframe suited to an use case. An aggregation significantly can reduce computational costs for obtaining inference results in DPGMs, but requires a careful consideration of indirect influences as [26] has shown. Under the assumption of an information processing chain it is reasonable to assume that during one timeframe no cyclic communication occurs, i.e., a feedback loop in information processing is not finished in one timeframe. Then, in fact, the presented DMIA model represents a so-called activator dynamic Bayesian network (ADBN) by Motzek et al. in [26] and [27] and one obtains well-defined semantics. Based on an ADBN, DMIAs can be reduced to, so-called, filtering and smoothing problems in ADBNs.

A reduction of DMIA to problems in (A)DBNs has significant advantages: One obtains the possibility to include *evidence* into a model, i.e., one is able to include information about actual observations of impacts on nodes. Then a *filtering* problem is formed by the problem of assessing the impact of any node at a time t given all obtained evidences so far, i.e., represents an evolution of impact "live" at the current time. Observations, e.g., can origin from IDS alerts, antivirus scans, battle reports or plain-sight observations. A significant advantage of (A)DBNs is that evidence is not only processed forwardly, but also backwardly. For example, if X is influenced by Y, and given an observed impact on node $X = {+x}$, an (A)DBN anticipates implications on Y by the observation of $X = {+x}$. Similarly, a *smoothing* problem is formed by an impact assessment of any node at time k, given evidence obtained until time t. A solution to a smoothing problem delivers valuable information for forensic analyses about intermediate states of mission impacts in retrospect, by including implications of future observations to preceding timeslices, i.e., preceding stages of impact assessments.

Moreover, considering a Markov-1 property in ADBNs allows for *persisting* impacts. This is beneficial for situations, where a potential compromise on a node X already has lead to an impact on other nodes $\vec{Y}$, whose inferred impact will persists if even, e.g., a cleaning operation is performed on the origin node X at some time t. A Markov-1 ADBN is able to raise awareness for a potential impact on nodes $\vec{Y}$ and therefore on a higher goal, such as a mission, even though an original source has been eliminated, but residues remain of them.

In order to deploy DMIA based on ADBN, one has to carefully consider the role of persistence, roles of external shock events and the roles of observations. In the following three paragraphs we discuss these roles and modeling approaches and outline potential applications of discussed models.

Persistence is the degree an impact shall persist over time, i.e., is a conditional probability fragment $p({+x^t}|{+x^{t-1}})$. In essence, two modeling approaches can be taken: $p({+x^t}|{+x^{t-1}}) = 1$ and $p({+x^t}|{+x^{t-1}}) < 1$. For the first option one obtains a model in which any potentially caused impact will persist forever, and will "aggregate" over time. Note that such an aggregation is significantly different from an aggregation obtained through a score-based propagation approaches, as one considers prior random probability distributions. For example, if there exists solely a prior random variable Q with $P({+q}) = 0.1$ as the only source of any impact in a network, every impact assessment will remain below 0.1. Moreover, irregardless how often any node came in contact with the potential impact source, all probabilities of impact will vanish, once one observes $\neg q$. We believe that it is highly complicated to include these mechanisms into any score-based propagation approach. A total persistence approach is suited to impact assessments for ICT networks, as resources usually are not able to "heal" themselves from any adversarially caused impacts. The second option, $p({+x^t}|{+x^{t-1}}) < 1$, resembles a domain in which impacts do not completely persists, and "cool down" over time. This is an approach suited to, e.g., infectious disease monitoring, where, in fact, entities modeled in a resource dependency model, i.e., people, actually heal themselves over time. It is arguable that a decaying impact is as well applicable to an ICT related use case, as a compromise will unlikely persist forever, as compromised data (in compromised memory locations and compromised data on a file system) will consecutively be exchanged.

Observations of impacts and non-impacts of random variables must be carefully differentiated in a dynamic MIA based on an ADBN. One has to differentiate between induced observations and true observations; a differentiation related to

Pearl's [28] introduction on the do-calculus and is best explained at an example: Considering, an infectious-disease monitoring system, one has to differentiate between a person being healthy, because he *has been* healed and between a person being tested to be healthy. The latter implies that there exists the probability that the person has never been infected, i.e., did never have a possibility to infect other persons he came in contact with. The first, is an induced action and delivers solely information from the current state on. In an ICT security related use case this means that if one actively cleans a system, the system is not impacted *from now on*, but previously caused impacts will persist. If one deeply and passively inspects a system and comes to the conclusion that the system has never been compromised, one truly observes $\neg x$, which has an effect on all previously caused transitive impacts. In order to model these differences, one needs to reconsider the role of external shock events.

External shock events in an ADBN used for DMIA are used to include soft evidence and external sources of observations. In order to include external sources of observations, one is able to model an external shock event that inevitably will cause the desired observation. For example, to observe a non-impacted node $\neg x^t$, but X has actively been cleaned at time t, one includes a random variable SE^t with $P(+se^t) = 1$ and $p(\neg x^t|+se^t) = 1$ s.t. $\forall \vec{z} : P(\neg x^t|+se^t, \vec{z}) = 1$. Due to a context-specific independence (cf. [29]) of X^t on all its possible dependencies $\vec{Z}$, given $+se^t$, X^t is decoupled from all other potential sources of (non-)impact, and the observation $\neg x^t$ is completely accredited to SE^t. Using external shock events to include soft evidence is especially useful to include information obtained from present vulnerabilities on nodes in an ICT security use case, which we discuss in the following example.

Example 6 (Roles of vulnerabilities and initial believes in DMIA)**.** *In an ICT/ICS security related use case of a dynamic mission impact assessment (DMIA), one must include information of potential compromises, i.e., information obtained through IDS alerts and from vulnerability advisories. An IDS alert, say "compromise on node X" represents a classic observation $+x^t$ and will affect other, transitively dependent devices even from previous timesteps, as the source of this observed compromise will lie in the network. Still, such sources must be modeled in an ADBN. We envision two potential use cases of DMIA, depending on two perspectives what should be achieved: (1) Viewing DMIA as a simulation of evolution of compromise, all sources of impacts are modeled as initial believes about compromise in R^0 in prior random distributions $P(X^0)$. From this perspective, all future observations are credited to any circumstance that is modeled, i.e., all impacts must origin from inside a network. In order to integrate a degree of flexibility, one is able to include a "leakiness" in CPDs, i.e., $P(+x^t|\neg\vec{z}^t) > 0$ ($\vec{z}^t$ representing all dependencies of X^t). Still, by modeling leaky CPDs, one loses deeper explanations for potential impacts, as impacts can occur "from nowhere." (2) Viewing DMIA as a monitoring system, one is able to explicitly model all potential sources of impacts at every timestep, but might over-accredit external- instead of internal-sources of impact. From a monitoring* *perspective, at every timestep for every vulnerability V an external shock event SE_V exists, as discussed in Example 4, and respective CPDs of affected nodes X accredit a (new) potential compromise due to this vulnerability at every timestep, i.e.,* $p(+x|+se_V) > 0$ *and* $P(+x|\neg\vec{z}) = 0$. *By following this approach, one is further able to represent that it is more likely that a node becomes compromised, given it is prone to a publicly known vulnerability and actively retrieves information from a dependent, impacted node.* ♦

This example outlines and discusses how to include information about (potential) compromises and threats in an ICT security DMIA based on an ADBN and outlines two possible perspectives that can be taken when using DMIA.

A deeper analysis and evaluation of DMIA is left for future work, as DMIA represents a too large novel research field and is beyond the scope of this article. Required probabilistic inference in ADBNs, as in DBNs, is a very hard problem, and approximate inference techniques are required to achieve linear scalability in large domains. Future work is dedicated to adapt existing approximation techniques, e.g., Rao-Blackwellised particle filtering as presented in [30] for DBNs towards novel ADBNs. We are certain that with approximate inference techniques for ADBNs, DMIA will be tractable even in large domains, and will scale linearly with the number of vertices in dependency models R^t and M^t.

7 Conclusion

We present a well-defined mathematical mission impact assessment, based on a probabilistic approach, without introducing score-based propagation algorithms returning spurious results.

We rely on the expertise of different experts and merge all views without losing information or forcing an expert into a knowledge field he cannot understand. All defined parameters are validatable and understandable locally, i.e., do not require a global view towards a complete system and algorithm. Based on an established mathematical model, we reduce mission impact assessment onto an well-understood problem in computer science. Experimental results demonstrate scalability of the approach such that large-scale network scenarios can be handed. We demonstrate the direct applicability of our approach in two real world use cases and present two possible applications of the presented impact assessment. Moreover, we discuss an extension of the presented approach towards dynamic mission impact assessments, providing a predictive and retrospect analysis of impacts over time, based on a novel dynamic probabilistic graphical model presented by [26].

Further future work is dedicated to integrating the presented mission impact assessment into a fully automated cyber defense system and to extend the work towards described time-dependent models. Examples 3 and 4 are closely linked to each other and motivate that an automated response system can be reduced to a mathematical minimization of the expected mission impact in our model in a predictive and retrospect analysis over time in changing, dynamic environments.

Acknowledgments

This work was partly supported by the Seventh Framework Programme (FP7) of the European Commission as part of the Panoptesec integrated research project (GA 610416).

We would like to thank the Swedish Defence Research Agency (FOI) for providing the SMIA2011 and SMIA2012 dataset, and especially Teodor Sommestad from the FOI's Informtion & Aeronautical Systems for further helpful comments and suggestions.

A Complexity Evaluation of MIA

The following figures substantiate Proof 4.2 on Theorem 1 about the approximation procedure by Definition 9 for obtaining an approximate and exact solution to the MIA problem by Definition 7.

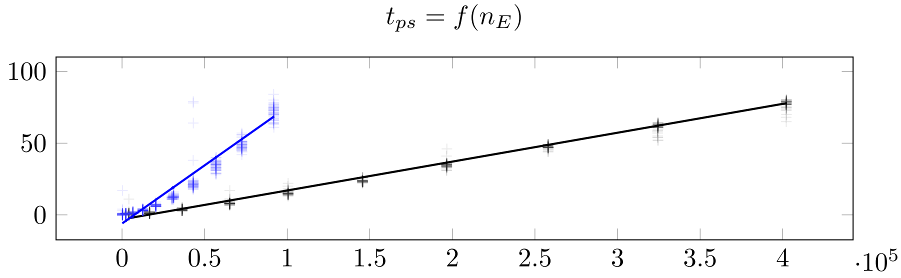

Figure 9: Proofsearch time (t_{ps} in ms, ordinate) is linear with the number of edges in the graph (n_e, abscissa). Black: $d_{max} = 3$, Blue: $d_{max} = 4$. n_N is linearly increased, meaning a quadratic increase of edges, i.e., $t_{ps} = f(n_E, d_{max}, n_{MA})$, but n_{MA} only contributes very slightly. 1000 datapoints from 50 experiments for $d_{max} = 3$ and 50 experiments for $d_{max} = 4$.

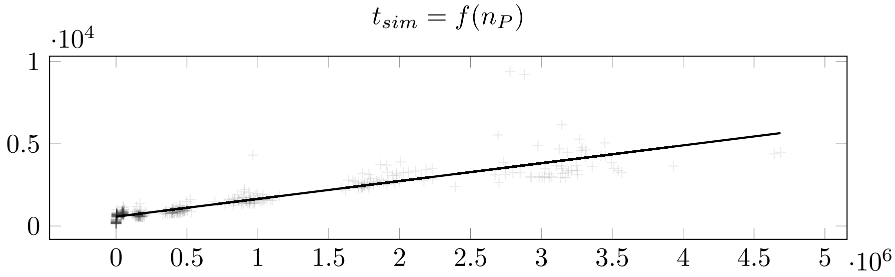

Figure 10: Simulation time (t_{sim} in ms, ordinate) is linear with the found proofs (abscissa). $n_S = 10000$, $d_{max} = 5$. n_N is linearly increased, meaning a linear increase of shock events and a quadratic increase of edges, both increasing the amount of proofs. Linear time complexity is also achieved for very large proof sets. 400 datapoints from 50 experiments.

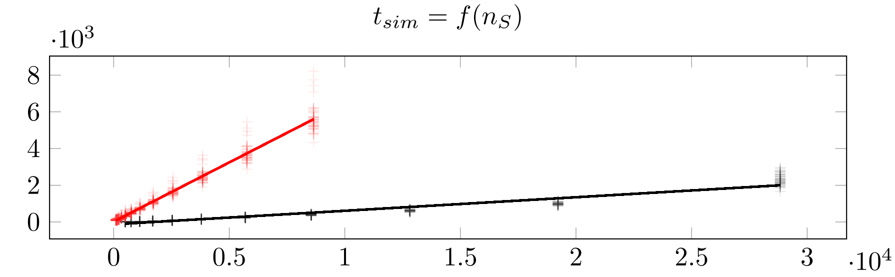

Figure 11: Simulation time (t_{sim} in ms, ordinate) directly correlates with the number of Monte Carlo simulations (abscissa). Unsimulatable proof cut-off (black) increases simulation time drastically (red without cut-off). $n_N = 500$, $n_{SE} = 50$, $d_{max} = 4$. n_S is increased exponentially and every measurement is repeated for 50 different random graphs, generating 1000 datapoints.

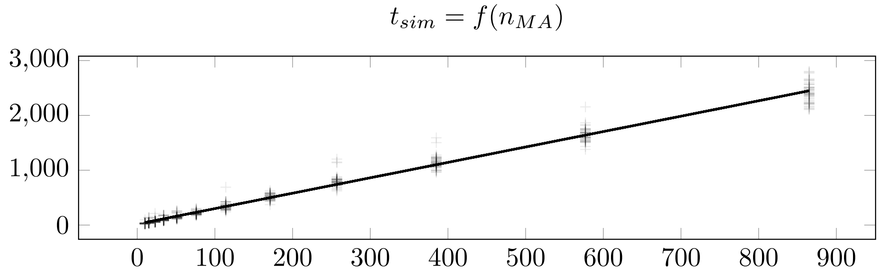

Figure 12: Simulation time (t_{sim} in ms, ordinate) directly correlates with the number of shock events (abscissa). Constant $n_N = 1000$, $d_{max} = 3$, $n_S = 1000$. n_{SE} is increased exponentially and repeated in 50 different random graphs, generating 600 datapoints.

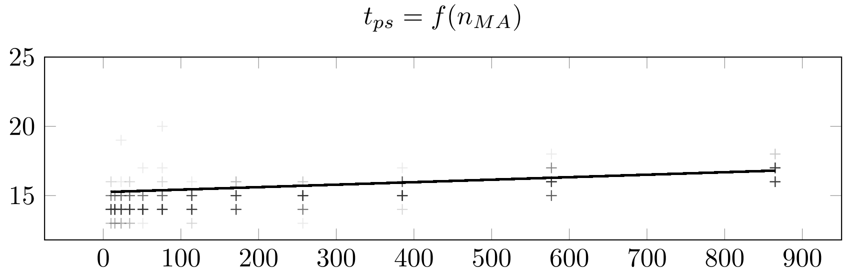

Figure 13: Proofsearch time (t_{ps} in ms, ordinate) is negligible dependent of the number of shock events (abscissa). t_{ps} correlates with the number of edges in the graph. Measurements collected during Figure 12.

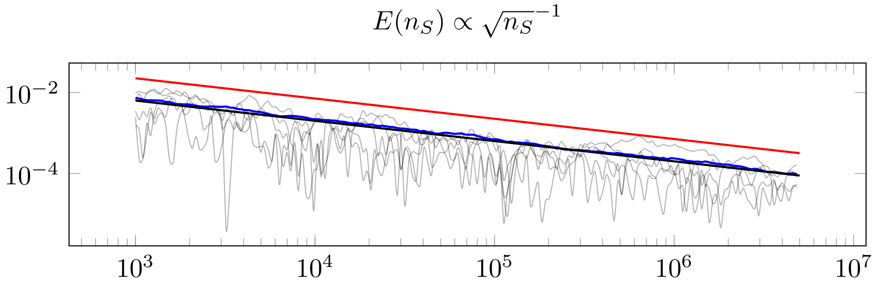

Figure 14: Absolute error (ordinate) of Monte Carlo approximation compared to an exact solution for large sampling domains (number of Monte Carlo samples, ordinate). Evaluated for one constant network, 100 times. Even for large sampling spaces the mean absolute error (blue) follows $\bar{E}(n_S) \approx 0.2\sqrt{n_S}^{-1}$ (black). Upper three-sigma bound displayed in red and five sample evaluations given for reference in gray. *Double logarithmic plot.*

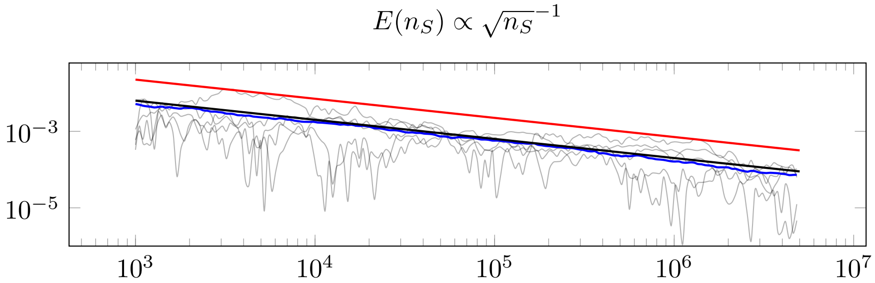

Figure 15: Absolute error (ordinate) of Monte Carlo approximation compared to an exact solution for large sampling domains (number of Monte Carlo samples, ordinate). Same evaluation as in Figure 14, but for 100 different, randomly generated problems. *Double logarithmic plot.*

References

References

[1] A. de Barros Barreto, P. C. G. da Costa, E. T. Yano, Using a Semantic Approach to Cyber Impact Assessment, in: STIDS 2013: 8th Conference on Semantic Technologies for Intelligence, Defense, and Security, Fairfax, Virginia, USA, November 12-15, 2013, pp. 101–108.

[2] M. Albanese, S. Jajodia, R. Jhawar, V. Piuri, Reliable Mission Deployment in Vulnerable Distributed Systems, in: DSN 2013: 43rd IEEE/IFIP In-

ternational Conference on Dependable Systems and Networks Workshop, Budapest, Hungary, June 24-27, IEEE, 2013, pp. 1–8.

[3] D. L. Buckshaw, G. S. Parnell, W. L. Unkenholz, D. L. Parks, J. M. Wallner, O. S. Saydjari, Mission Oriented Risk and Design Analysis of Critical Information Systems, Military Operations Research 10 (2) (2005) 19–38.

[4] G. Jakobson, Mission Cyber Security Situation Assessment using Impact Dependency Graphs, in: FUSION 2011: 14th International Conference on Information Fusion, Chicago, Illinois, USA, July 5-8, 2011, pp. 1–8.

[5] S. Musman, A. Temin, M. Tanner, D. Fox, B. Pridemore, Evaluating the Impact of Cyber Attacks on Missions, in: ICIW 2010: 5th International Conference on Information Warfare and Security, Wright-Patterson Air Force Base, Ohio, USA, April 8-9, 2010, pp. 446–456.

[6] J. R. Goodall, A. D'Amico, J. K. Kopylec, Camus: Automatically Mapping Cyber Assets to Missions and Users, in: MILCOM 2009: IEEE Military Communications Conference, Boston, Massachusetts, USA, October 18-21, IEEE, 2009, pp. 1–7.

[7] A. D'Amico, L. Buchanan, J. Goodall, P. Walczak, Mission Impact of Cyber Events: Scenarios and Ontology to Express the Relationships between Cyber Assets, Missions, and Users, in: ICIW 2010: 5th International Conference on Information Warfare and Security, Wright-Patterson Air Force Base, Ohio, USA, April 8-9, 2010, pp. 8–9.

[8] L. Wang, T. Islam, T. Long, A. Singhal, S. Jajodia, An Attack Graph-Based Probabilistic Security Metric, in: DBSec 2008: 22nd Annual IFIP WG 11.3 Conference on Data and Applications Security, London, UK, July 13-16, 2008, pp. 283–296.

[9] Y. Liu, H. Man, Network Vulnerability Assessment Using Bayesian Networks, in: SPIE Vol. 5812: Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security, Orlando, Florida, USA, March 28, International Society for Optics and Photonics, 2005, pp. 61–71.

[10] P. Xie, J. H. Li, X. Ou, P. Liu, R. Levy, Using Bayesian Networks for Cyber Security Analysis, in: DSN 2010: 40th IEEE/IFIP International Conference on Dependable Systems and Networks 2010, Chicago, Illinois, USA, June 28 - July 1, 2010, pp. 211–220.

[11] N. Kheir, H. Debar, N. Cuppens-Boulahia, F. Cuppens, J. Viinikka, Cost Evaluation for Intrusion Response Using Dependency Graphs, in: N2S 2009: International Conference on Network and Service Security, Paris, France, June 24-26, IEEE, 2009, pp. 1–6.

[12] M. Jahnke, C. Thul, P. Martini, Graph based Metrics for Intrusion Response Measures in Computer Networks, in: LCN 2007: 32nd Annual IEEE Conference on Local Computer Networks, Clontarf Castle, Dublin, Ireland, October 15-18, 2007, pp. 1035–1042.

[13] M. Henrion, Practical Issues in Constructing a Bayes Belief Network, International Journal of Approximate Reasoning 2 (3) (1988) 337.

[14] J. Pearl, S. Russell, Bayesian Networks, in: M. A. Arbib (Ed.), Handbook of Brain Theory and Neural Networks, MIT Press, 2003, pp. 157–160.

[15] A. Motzek, C. Geick, R. Möller, Semantic Normalization and Merging of Business Dependency Models, in: CBI 2016: 18th IEEE Conference on Business Informatics, Paris, France, August 29 - September 1, 2016, pp. in–press.

[16] S. Jha, O. Sheyner, J. Wing, Two Formal Analyses of Attack Graphs, in: CSFW 2002: 15th IEEE Workshop on Computer Security Foundations, Cape Breton, Nova Scotia, Canada, June 24-26, IEEE, 2002, pp. 49–63.

[17] X. Ou, S. Govindavajhala, A. W. Appel, MulVAL: A Logic-based Network Security Analyzer, in: 14th USENIX Security Symposium, Baltimore, Maryland, USA, July 31 - August 5, 2005.

[18] R. Langner, To Kill a Centrifuge. A Technical Analysis of What Stuxnet's Creators Tried to Achieve, Tech. rep. (2013).

[19] L. D. Raedt, A. Kimmig, H. Toivonen, ProbLog: A Probabilistic Prolog and Its Application in Link Discovery, in: IJCAI 2007: 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007, pp. 2462–2467.

[20] A. B. Owen, Monte Carlo Theory, Methods and Examples, 2013.

[21] T. Sommestad, A. Hunstad, Intrusion Detection and the Role of the System Administrator, Information Management & Computer Security 21 (1) (2013) 30–40.

[22] M. Bastian, S. Heymann, M. Jacomy, Gephi: An Open Source Software for Exploring and Manipulating Networks, in: ICWSM 2009: 3rd International Conference on Weblogs and Social Media, San Jose, California, USA, May 17-20, 2009.

[23] Panoptesec DOW, Panoptesec Annex I, Description Of Work, in: Project Deliverables of the Panoptesec Collaborative Research Project on Dynamic Risk Approaches for Automated Cyber Defence, Grant Agreement No: 610416, ICT-2013.1.5, Trustworthy ICT, (Version September 4th, 2015), 2013.

[24] Acea SpA, The Acea Group (2016) [cited 06.05.2016].
URL http://www.acea.it/section.aspx/en/acea_spa

[25] G. G. Granadillo, A. Motzek, J. Garcia-Alfaro, H. Debar, Selection of Mitigation Actions Based on Financial and Operational Impact Assessments, in: ARES 2016: 11th International Conference on Availability, Reliability and Security, Salzburg, Austria, August 31 - September 2, 2016, pp. in–press.

[26] A. Motzek, R. Möller, Indirect Causes in Dynamic Bayesian Networks Revisited, in: IJCAI 2015: 24th International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, July 25-31, 2015, pp. 703–709.

[27] A. Motzek, R. Möller, Exploiting Innocuousness in Bayesian Networks, in: AI 2015: 28th Australasian Joint Conference on Artificial Intelligence, Canberra, ACT, Australia, November 30 - December 4, 2015, pp. 411–423.

[28] J. Pearl, Reasoning with Cause and Effect, AI Magazine 23 (1) (2002) 95–112.

[29] C. Boutilier, N. Friedman, M. Goldszmidt, D. Koller, Context-Specific Independence in Bayesian Networks, in: UAI 1996: 12th Conference on Uncertainty in Artificial Intelligence, Reed College, Portland, Oregon, USA, August 1-4, 1996, pp. 115–123.

[30] A. Doucet, N. de Freitas, K. P. Murphy, S. J. Russell, Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks, in: UAI 2000: 16th Conference on Uncertainty in Artificial Intelligence, Stanford University, Stanford, California, USA, June 30 - July 3, 2000, pp. 176–183.